

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**SYSTÉM PRO VZDÁLENÝ MONITORING A ŘÍZENÍ  
PROCESU VÝROBY PIVA**

SYSTEM FOR REMOTE MONITORING OF BEER BREWING PROCESS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Oldřich Klíma**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Ing. Jiří Hošek, Ph.D.**

**BRNO 2017**



# Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

**Student:** Oldřich Klíma

**ID:** 164903

**Ročník:** 3

**Akademický rok:** 2016/17

## NÁZEV TÉMATU:

### **Systém pro vzdálený monitoring a řízení procesu výroby piva**

## POKYNY PRO VYPRACOVÁNÍ:

Bakalářská práce se bude zabývat problematikou automatizovaného řízení procesu výroby piva s využitím embedded zařazení Raspberry Pi. Cílem bude navrhnout a zprovoznit systém složený z řídicí jednotky, inteligentního přepínače (ideálně bezdrátového), teplotního senzoru a varné nádoby. Jako řídicí software bude použita open-source platforma CraftBeerPi, která bude v případě potřeby upravena tak, aby splňovala požadavky vytvořeného systému. Řídicí jednotka bude na základě definovaných pravidel a aktuální teploty ovládat spínání varné nádoby přes inteligentní přepínač. Návrh i implementace systému budou podrobně zdokumentovány.

## DOPORUČENÁ LITERATURA:

[1] GUBBI, Jayavardhana, et al. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 2013, 29.7: 1645-1660.

[2] MONK, Simon. 2013. Programming the Raspberry Pi: getting started with Python. New York: McGraw-Hill. ISBN 00-718-0783-7.

**Termín zadání:** 1.2.2017

**Termín odevzdání:** 8.6.2017

**Vedoucí práce:** doc. Ing. Jiří Hošek, Ph.D.

**Konzultant:**

**doc. Ing. Jiří Mišurec, CSc.**  
předseda oborové rady

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Bakalářská práce se zabývá systémy s otevřeným (volně dostupným) zdrojovým kódem pro vzdálený monitoring a řízení procesu výroby piva. V první části dochází k seznámení s problematikou vaření piva. Je zde popsán základní postup výroby piva a vytyčeny zásadní rozdíly mezi výrobou v průmyslovém pivovaru, minipivovaru nebo v domácích podmínkách. Následuje rozbor problematiky Internetu věcí a výběr vhodného embedded zařízení, na kterém bude zprovozněn jeden z několika porovnávaných řídicích programů. Dále dochází k seznámení s ostatním hardwarem, potřebným pro automatizaci důležitých procesů v pivovarnictví a nastínění vybavení jednoduché automatizované varny. Dochází též k popisu dvou základních modelů vzorové varny řízené klasicky nebo bezdrátově. Nakonec je popsán postup implementace zvoleného systému a přiblížen způsob úpravy aplikace pro zajištění bezproblémové funkčnosti a bezdrátovou komunikaci s ovládanými spínači a senzory. Poslední kapitola má za cíl seznámit čtenáře s postupem nastavení a ovládání vytvořeného programového řešení.

## KLÍČOVÁ SLOVA

Raspberry Pi, Internet věcí, vestavěný systém, jednodeskový počítač, webový server, WiFi, protokol MQTT

## ABSTRACT

Bachelor's thesis is focused on problematics of remote monitoring and process control system for homebrewing, that are based on open source software. In the first chapter there is an introduction into beer brewing processes and the differences between industrial, small and home brewery is described. In the next chapter there is an information about Internet of Things and suitable embedded platform is choosen. One of the compared software solution will be instaleand on this platform. The procedure of implementation is described in the next parts and thesis is then focusing on changes in the program of backend server application. These changes are improving stability of whole system and they are adding support for wireless communication with controlled switches and sensors. Last part of this thesis introduces reader into set up and controll process of the software solution.

## KEYWORDS

Raspberry Pi, Internet of Things, embedded system, single-board computer, web server, WiFi, MQTT protocol

KLÍMA, Oldřich *Systém pro vzdálený monitoring a řízení procesu výroby piva*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 66 s. Vedoucí práce byl doc. Ing. Jiří Hošek, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Systém pro vzdálený monitoring a řízení procesu výroby piva“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Jiřímu Hoškovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....  
podpis autora

# OBSAH

<b>Úvod</b>	<b>12</b>
<b>1 Proces výroby piva</b>	<b>13</b>
1.1 Výroba v průmyslovém pivovaru . . . . .	13
1.2 Výroba v domácích podmínkách . . . . .	14
<b>2 Výběr vhodného systému pro vzdálený monitoring</b>	<b>16</b>
2.1 Nejznámější platformy pro Internet věcí . . . . .	16
2.1.1 Arduino . . . . .	16
2.1.2 Raspberry Pi . . . . .	17
2.1.3 Banana Pi . . . . .	18
2.1.4 Srovnání a výběr platformy . . . . .	18
2.2 Systémy s otevřeným zdrojovým kódem pro vzdálený monitoring . . .	19
2.2.1 BrewBot . . . . .	19
2.2.2 BrewPi . . . . .	20
2.2.3 CraftBeerPi . . . . .	20
2.2.4 MashBerry . . . . .	21
2.2.5 Vybraný program pro vzdálený monitoring . . . . .	22
<b>3 Návrhy automatizované varny</b>	<b>23</b>
3.1 Varné nádoby . . . . .	23
3.1.1 Úprava pro vystírání . . . . .	23
3.1.2 Chmelovar . . . . .	24
3.1.3 Nádoba na horkou vodu . . . . .	24
3.1.4 Čerpadla . . . . .	24
3.2 Elektronika systému s uspořádáním RIMS . . . . .	25
3.2.1 Řídící jednotka . . . . .	25
3.2.2 Měření teploty . . . . .	26
3.2.3 Výkonové spínání . . . . .	26
3.3 Elektronika systému s uspořádáním RIMS rozšířený o bezdrátovou komunikaci . . . . .	27
3.3.1 Řídící jednotka . . . . .	28
3.3.2 Bezdrátové spínání . . . . .	28
3.3.3 Bezdrátová komunikace s měřícím čidlem . . . . .	28
<b>4 Implementace softwaru pro vzdálený monitoring</b>	<b>30</b>
4.1 Instalace operačního systému Raspbian . . . . .	30
4.2 Instalace CraftBeerPi . . . . .	30



4.3	Webová aplikace . . . . .	31
4.3.1	První spuštění . . . . .	32
4.3.2	Úvodní obrazovka . . . . .	33
4.3.3	Fermentace . . . . .	33
4.3.4	Správa kroků a receptů . . . . .	34
4.3.5	Ostatní funkce . . . . .	34
4.4	Úpravy aplikace CraftBeerPi pro řízení bezdrátových spínačů . . . . .	35
4.4.1	Protokol MQTT a jeho implementace do CraftBeerPi . . . . .	36
4.4.2	Vytvoření seznamu MQTT klientů . . . . .	37
4.4.3	Řízení vstupně/výstupních portů protokolem MQTT . . . . .	38
4.4.4	Získání teploty z MQTT zprávy . . . . .	38
4.4.5	Implementace do webového rozhraní . . . . .	38
<b>5</b>	<b>Připojení spínačů a senzorů</b>	
	<b>k Raspberry Pi</b>	<b>40</b>
5.1	Vytváření bezdrátového spojení . . . . .	41
5.2	Bezdrátový mikrokontroler ESP8266 . . . . .	42
5.2.1	Výběr vhodného řídicího programu pro bezdrátový mikrokon- troler ESP8266 . . . . .	43
5.2.2	Instalace a nastavení řídicího programu ESP Easy . . . . .	45
5.3	Shrnutí a rekapitulace instalace a úprav systému CraftBeerPi . . . . .	47
<b>6</b>	<b>Návod k použití autorem upraveného systému CrafterBeerPi</b>	<b>49</b>
6.1	Správa receptů . . . . .	50
6.2	Zahájení a průběh procesu vaření piva . . . . .	51
	<b>Literatura</b>	<b>54</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>57</b>
	<b>Seznam příloh</b>	<b>59</b>
<b>A</b>	<b>Příloha – Vlastní zdrojové kódy</b>	<b>60</b>
A.1	Program pro získání seznamu připojených bezdrátových zařízení . . . . .	60
A.2	Program pro bezdrátové řízení výstupů přes MQTT protokol . . . . .	62
A.3	Program pro čtení teploty z MQTT zpráv . . . . .	64
<b>B</b>	<b>Příloha – Obsah přiloženého CD</b>	<b>66</b>

# SEZNAM OBRÁZKŮ

1.1	Varna Novoměstského pivovaru[2]	14
1.2	Příklad domácího vaření piva[4]	15
2.1	Ovládací skříň systému BrewBot, tzv. Control Box [9]	19
2.2	Webová stránka ovládacího rozhraní CraftBeerPi[12]	20
2.3	Řídící jednotka systému MashBerry [13]	21
3.1	Varná souprava v konfiguraci HERMS	24
3.2	Znázorněný příklad vzorového uspořádání systému typu RIMS	25
3.3	Vodotěsné teplotní čidlo DS1820 [17]	26
3.4	Znázornění systému RIMS, který komunikuje bezdrátově	27
3.5	Bezdrátový spínač Sonoff [19]	28
3.6	Bezdrátový spínač Sonoff TH [20]	29
4.1	Úvodní obrazovka instalačního skriptu	31
4.2	Server, běžící v příkazové řádce	32
4.3	Nastavení systému	33
4.4	Úvodní stránka CraftBeerPi	34
4.5	Nastavení varné nádoby	35
4.6	Příklad funkce protokolu MQTT	36
4.7	Úvodní nastavení systému po úpravách	39
5.1	Nejjednodušší zapojení hardwaru CraftBeerPi [27]	40
5.2	Bezdrátový mikrokontroler ESP8266	43
5.3	Řídící program ESPurna pro ESP8266	44
5.4	Příklad připojených zařízení přes ESP Easy	45
5.5	Nastavení spínače Sonoff01	46
5.6	Nastavení připojeného senzoru DS18B20	47
6.1	Příklad nastavení vzorového systému pro vzdálený monitoring	49
6.2	Správce receptů programu CraftBeerpi	50
6.3	Vytvoření nového kroku v seznamu receptů	51

# SEZNAM TABULEK

2.1	Tabulka srovnání některých zařízení z rodiny Arduino[6]	17
2.2	Tabulka srovnání jednotlivých verzí počítače Raspberry Pi[7]	17
2.3	Tabulka srovnání některých používaných IoT platforem	18
2.4	Tabulka srovnání zvolených programů pro řízení procesu výroby piva	22
3.1	Tabulka vlastností spínačů Sonoff	29

# SEZNAM VÝPISŮ

5.1	Použitá konfigurace DHCP serveru . . . . .	41
A.1	Soubor, spravující komunikaci přes protokol MQTT a generování seznamu připojených zařízení . . . . .	60
A.2	Soubor pro řízení vstupně výstupních portů zařízení ESP8266 bezdrátovým přenosem informací . . . . .	62
A.3	Program na vyčtení informace o teplotě ze zachycené MQTT zprávy .	64

# ÚVOD

Jedním z velkých trendů dnešní doby je fenomén IoT (Internet of Things – Internet věcí). Internet věcí je označení pro různá zařízení, připojená k síti internet, která zprostředkovávají data uživatelům nebo sobě navzájem. Díky této technologii je možné na dálku zapnout domácí spotřebič, zkontrolovat postup práce při 3D tisku, nebo např. sledovat a měnit parametry zabezpečení objektu. To vše lze realizovat pohodlně například přes aplikaci v chytrém telefonu, tabletu nebo osobním počítači (PC), popřípadě přes webovou stránku.

Tato bakalářská práce se věnuje problematice vzdáleného monitoringu a řízení procesu výroby piva. Srovnává jednotlivé dostupné systémy a jejím výstupem je implementace vybraného programového řešení s otevřeným zdrojovým kódem na zvolené embedded zařízení. Výhodou volně dostupných aplikací je přístupnost jednotlivých prvků a zdrojových kódů, které lze dle potřeby volně upravovat a šířit. Ve většině případů jsou systémy sdílené pod licencí s otevřeným zdrojovým kódem doplněny početnou komunitou uživatelů, kteří jsou zdrojem cenných rad a informací.

Pivo je nápoj, který byl lidstvu znám už odpradávná. Za dobu jeho existence bylo vyvinuto mnoho různých způsobů přípravy a díky tomu i spousta jednotlivých pivních stylů. V současné době převládá výroba v průmyslových pivovarech. Velké oblibě se však těší i výroba v minipivovarech a domácí vaření piva. Automatizací některých potřebných procesů v pivovarnictví se tato práce zabývá. Díky ní bude možné nejen celý postup výroby provést pohodlněji, ale například i udržet stále stejný standard kvality, což je největší problém vaření v domácích podmínkách. Součástí práce je též několik návrhů jednoduchého automatizovaného systému a vhodná úprava vybrané volně dostupné řídicí aplikace pro zajištění co nejlepší funkčnosti.

# 1 PROCES VÝROBY PIVA

Proces výroby piva začíná vhodným výběrem čtyř základních surovin – vody, sladu, chmele a pivovarských kvasinek. Různé druhy sladu ovlivňují barvu (tmavé, světlé a polotmavé) a výslednou chuť. Jeho množství ovlivňuje stupňovitost (nyní EPM – Extrakt původní mladiny) výrobku. Množstvím chmele a jeho odrůdou se dá určit konečná hořkost. Pivovarské kvasinky se volí podle požadovaného pivního stylu. Svrchní kvašení je používáno například pro pivní styly Ale, Porter, Stout a mnoho dalších. Typickým představitelem spodního kvašení jsou piva plzeňského nebo vídeňského typu. Funkčně lze výrobu rozdělit do tří základních celků: vaření, kvašení a dozrávání. Při vaření je cílem získat ze vstupních surovin mladinu bohatou na zkvasitelné cukry. Ve fázi kvašení se cukry pomocí kvasinek přeměňují na alkohol. Dozrávání nejčastěji probíhá v tzv. ležáckém sklepě za předem dané teploty a ovlivňuje konečnou chuť a plnost piva.

## 1.1 Výroba v průmyslovém pivovaru

Průmyslová výroba piva probíhá ve varně pivovaru (např. na Obr. 1.1) a skládá se z několika procedur:

- Nejprve dochází ke **šrotování** sladu. Díky rozdrčení zrn dochází k lepšímu uvolnění obsažených látek. Slupky ze zrn (tzv. pluchy) v dalších částech výroby pomáhají při scezování.
- Slad je poté vsypáván do **vystírací kádě**, kde je promíchán s předem určenou dávkou vody (ovlivňuje se stupňovitost – více vody znamená menší hodnotu EPM). Tato procedura se nazývá **vystírání**.
- Následuje **rmutování**, při kterém dochází ke štěpení látek obsažených ve sladu na zkvasitelné cukry. Existují dvě základní postupy rmutování. Při **Infúzním rmutování** se postupně v určených intervalech dílo ohřívá na zadané teploty až do bodu varu. Při **Dekokčním rmutování**, které je používáno pro výrobu piva plzeňského typu, se zpravidla jedna třetina objemu převede do **rmutovací pánve** a je přivedena k varu. Následně se přečerpá zpět do vystírací pánve. Celý proces je několikrát opakován (nejčastěji dvakrát).
- Várka je poté scezena a vzniká tzv. **sladina**. Jedná se o velmi sladký roztok, který obsahuje zkvasitelné cukry a mnoho dalších látek.
- Sladina je přečerpána do **mladinové pánve** a přivedena k bodu varu. Následuje **chmelení**, při kterém se určuje výsledná hořkost piva a vzniká **mladina**. V České republice se chmelí nejčastěji třikrát. Objem chmele je rozdělen na čtvrtiny. První se vsypává na začátku vaření. Další dvě čtvrtiny se přidávají

těsně před dosažením bodu varu a poslední čtvrtina se použije před ukončením vaření (většinou do 10 minut) a na hořkost má největší vliv.

- Aby pivo dobře zkvasilo je třeba mladinu prudce zchladit na zákvasnou teplotu (liší se podle kmene použitých kvasinek). Poslední fází je **kvašení** a **dozrávání**. Délkou kvašení se určuje podíl zbytkových cukrů a obsah alkoholu. Dozrávání v ležáckém sklepě dotváří charakter výsledného produktu. Doba ležení se pohybuje v řádu týdnů [1].



Obr. 1.1: Varna Novoměstského pivovaru[2]

## 1.2 Výroba v domácích podmínkách

Výroba piva v domácích podmínkách se od výroby průmyslové značně odlišuje. Je to dáno podstatně menším objemem<sup>1</sup> výroby. Díky tomu bývá občas celý proces zjednodušen. Jednotlivé body výroby zůstávají stejné, jako je popsáno v části 1.1, velmi často však navazují jeden na druhý ve stejné varné nádobě (viz Obr.1.2). V nejjednodušší variantě je použit pouze jeden kotel, do kterého se postupně přidávají všechny suroviny a zahříváním na nastavenou teplotu dochází nejprve ke vystírání, scezování (většinou se provádí přečerpáním obsahu přes síto), Infúznímu rmutování a nakonec je přidán chmel. Pokud se objem várky pohybuje v řádu litrů, stačí k realizaci pouze průměrně vybavená domácí kuchyně. V zahraniční literatuře se však často setkáváme se složitějšími konstrukcemi domácí varny. Mezi dvě nej-používanější patří systém HERMS (Heat Exchanged Recirculation Mash System – Recirkulační vystírací/rmutovací systém s teplotní výměnou a RIMS (Recirculation

<sup>1</sup>Průmyslové pivovary zpracovávají objemy v řádu hektolitrů.

Infusion Mash System – Recirkulační Infúzní vystírací/rmutovací systém). Ve variantě HERMS je varna složena ze tří nádob, kdy první slouží k předehtování vody, která má být dále zužitkována v průběhu procesu vaření. Touto nádobou prochází recirkulační potrubí sladiny ze druhé (vystírací) nádoby. Tím odpadá potřeba mít v této nádobě zabudovaný vyhřívací systém neboť se změnou teploty v nádobě na horkou vodu se pozvolna změní i teplota sladiny. Třetí nádoba slouží jako chmelovar. Systém RIMS je naopak redukován pouze na dvě nádoby kdy jedna slouží opět pro vystírání a rmutování a druhá jako chmelovar. Uspořádání HERMS je oproti RIMS pokročilejší. Uživatel má mnohem lepší kontrolu nad teplotou várky, její změny jsou mnohem pozvolnější a při doplňování objemu odpařené vody do systému nedochází k teplotnímu šoku a zchlazení sladiny. Nicméně konstrukce je mnohem složitější, nákladnější a pro menší objemy várky spíše předimenzovaná.

Existuje ještě možnost výroby piva z předem připraveného mladinového, nebo sladinnového extraktu. Jedná se o volně dostupné koncentráty, dodávané na trh převážně velkými pivovary. Použitím koncentrátu je celý proces velmi zjednodušen, protože příprava mladiny je technicky, časově a energeticky nejnáročnější z celého postupu výroby piva. Tuto možnost využívají zejména nadšenci, kteří nemají prostředky pro zprovoznění vlastní varné soupravy, nebo začátečníci, jenž se s postupem výroby piva teprve seznamují [3]. Pivo, uvařené v domácích podmínkách, je velmi zajímavá alternativa k průmyslově unifikovaným komerčním výrobkům.



Obr. 1.2: Příklad domácího vaření piva[4]



## 2 VÝBĚR VHODNÉHO SYSTÉMU PRO VZDÁLENÝ MONITORING

Na systémy pro vzdálený monitoring a obecně na automatizované systémy se dá pohlížet z několika rovin. Jeden pohled zahrnuje hardwarovou úroveň, další rovina se je softwarová. Výběr obou je značně důležitý, neboť na něm závisí funkčnost celku. Nedostatečně dimenzovaný hardware nebude nikdy poskytovat dostatečnou hloubku automatizace a špatně navržený software zase nevyužije plný hardwarový potenciál. Poslední důležitý pohled je čistě praktický. Je nezbytné určit, kolik prostředků lze do automatizace investovat.

Vhodný systém pro vzdálený monitoring a řízení procesu výroby piva musí splňovat několik kritérií. Jeho software musí být dostatečně univerzální, aby jej bylo možné použít pro různé konfigurace varné sestavy. Musí umět získat námi požadované parametry jako např. teplotu, objem vody a mnoho dalších a v neposlední řadě je vhodně ovládat a interpretovat.

### 2.1 Nejznámější platformy pro Internet věcí

Platformy vhodné pro aplikaci v systémech Internet of Things – Internet věcí (IoT) zprostředkovávají interakci mezi uživatelem a zařízením skrze síť Internet. Tzn. musí mít přístup k Internetu (přes WiFi, GSM, Ethernet, atd.). Pro účely této bakalářské práce by měly mít dostatečný počet I/O (Input/Output – vstupně výstupní výstupů, kvůli ovládání jednotlivých prvků systému a dostatečný výkon pro hladký běh řídicího programu. Těmto podmínkám nejlépe vyhovují jednodeskové počítače, které jsou jednoduché na ovládání, mají dobrou konektivitu a v současné době jsou velmi dostupné. Mezi nejoblíbenější jednodeskové počítače patří výrobky firem Arduino a Raspberry Pi Foundation. Některé<sup>1</sup> jejich modely budou dále specifikovány [5].

#### 2.1.1 Arduino

Arduino je jednodeskový počítač založený na 8 bitových mikrokontrolérech ATmega (Arduino Due obsahuje 32 bitový ARM procesor). Jedná se o otevřenou platformu s grafickým vývojovým prostředím, která se dá snadno rozšířit pomocí rozšiřujících obvodů (tzv. Shieldů. Např. Ethernet shield, senzor shield. . . ). Řídicí program je vyvíjen zvlášť a do zařízení je posléze nahrán a spuštěn, kdy běží v nekonečné smyčce

---

<sup>1</sup>Autor vybíral z modelů, které jsou oblíbené mezi běžnými uživateli a jsou schopné podat dostatečný výkon pro zprovoznění systému vzdáleného monitoringu.

a reaguje na podněty okolí. Nejpoužívanějším modelem je Arduino Uno s mikrokontrolérem ATmega328. Základní deska zařízení má většinu I/O pinů přístupných přes standardizované patice, do kterých se jednoduše připojují další obvody, nebo Shiedly (14 I/O digitálních pinů a 6 pinů analogových se softwarově řízeným PWM (Pulse Width Modulation – Pulzně šířková modulace) výstupem).[6] Díky otevřenosti celého systému vznikla celá řada klonů, na kterých se podílejí různé firmy i jednotlivci (např. Funduino, Freeduino, LABduino...). Srovnání některých výrobků lze nalézt v Tab. 2.1.

Tab. 2.1: Tabulka srovnání některých zařízení z rodiny Arduino[6]

Model	Procesor	Takt	Analog I/O	IO/PWM	RAM
Uno	ATmega328P	16 MHz	6/0	14/6	2 kB
Due	ATSAM3X8E	84 MHz	12/2	54/12	96 kB
Zero	ATSAMD21G18	48 MHz	6/1	14/10	32 kB

### 2.1.2 Raspberry Pi

Raspberry Pi je jednodeskový počítač, vytvořený v roce 2012 britskou Raspberry Pi Foundation. Jeho popularita dala vzniknout velkému množství klonů (viz 2.1.3). Existuje v několika verzích, které se mezi sebou liší dostupným výpočetním výkonem a cenou. Primárním operačním systémem jsou distribuce Linuxu. Na základní desce nachází procesory z rodiny ARM Cortex v různých provedeních a taktu, závislejícím na zvoleném modelu (porovnání v Tab. 2.2). K dispozici má většinou 1GB paměti RAM. O připojení periférií se stará dvojice, nebo i čtveřice USB 2.0 portů, HDMI výstup a mnoho dalších. Pro přístup k internetu je na desce 10/100 Mbps konektor RJ-45. Komunikaci s I/O prvky zajišťuje GPIO sběrnice. Raspberry Pi model 3 B umožňuje navíc bezdrátovou komunikaci přes WiFi (802.11 b/g/n) a Bluetooth (BT) verze 4.1. Operační systém se zavádí z připojené MicroSD karty a na výběr máme z mnoha podporovaných distribucí Linuxu (Raspbian, Pidora a další...) [7].

Tab. 2.2: Tabulka srovnání jednotlivých verzí počítače Raspberry Pi[7]

Model	Procesor	Takt	RAM	USB	Proud [mA]
1 B	ARM1176JZF-S	700 MHz	512 kB	2x2.0	400
2 B	ARM Cortex-A7 quad	900 MHz	1024 kB	4x2.0	310
Zero	BCM 2835 SOC	1 GHz	512 kB	1x2.0M	140
3 B	ARMv8 quad 64bit	1,2 GHz	1024 kB	4x2.0M	310

### 2.1.3 Banana Pi

Banana Pi je klon proslulého systému Raspberry Pi od Raspberry Pi Foundation. Jedná se o jednodeskový počítač vyráběný firmou Shenzhen SINOVOIP Co. Ltd. existující v různých verzích a hardwarových modifikacích. Základem Banana Pi je dvoujádrový (nebo čtyřjádrový procesor ARM Cortex-A7 s 1GB DDR3 pamětí RAM. Pro připojení periferií nabízí 2 USB 2.0 porty, pro I/O komunikaci je určena sběrnice GPIO. Grafický výstup umožňuje GPU ARM Mali400MP, zobrazovací zařízení je možné připojit přes několik portů (HDMI, CVBS, LVDS/RGB). Zařízení nemá interní úložiště a operační systém je zaváděn z připojené MicroSD karty. Pro Banana Pi M1 existuje několik dostupných operačních systémů a záleží pouze na potřebách uživatele, který z nich si vybere (např. Android 4.2, Raspbian, Ubuntu a další). Možnost připojit periferie jako myš, klávesnici a monitor z něj dělá téměř plnohodnotný počítač. Zařízení má též početnou komunitu uživatelů [8].

### 2.1.4 Srovnání a výběr platformy

Pro bakalářskou práci vybral autor ze zmíněných zařízení Raspberry Pi 3 model B. Jeho největší předností je, že všechny dále popsané aplikace pro vzdálený monitoring využívají jako platformu právě Raspberry. Arduino neposkytuje dostatečný výkon pro provoz stabilního operačního systému a bylo by nutné vyvinout kompletní aplikaci včetně implementace webového serveru. Banana Pi má podobné výkonové parametry jako Raspberry (viz srovnání v Tab. 2.3). Poskytuje navíc několik možností připojení hardwaru (SATA, infraport, atd.). Balíčky jednotlivých řešení však nejsou na této platformě odzkoušeny. Raspberry Pi 1 model B by nemuselo poskytnout dostatečný výkon pro hladký běh aplikace. Jeho nevýhodou je i vyšší příkon. Model 3 se 64 bitovou architekturou naopak poskytuje výkonu nadbytek. Díky integrovanému WiFi modulu ovšem odpadají problémy s kompatibilitou ovladačů USB WiFi (Standardy IEEE 802.11 pro připojení k bezdrátovým lokálním sítím) karet a operačního systému Raspbian.

Tab. 2.3: Tabulka srovnání některých používaných IoT platforem

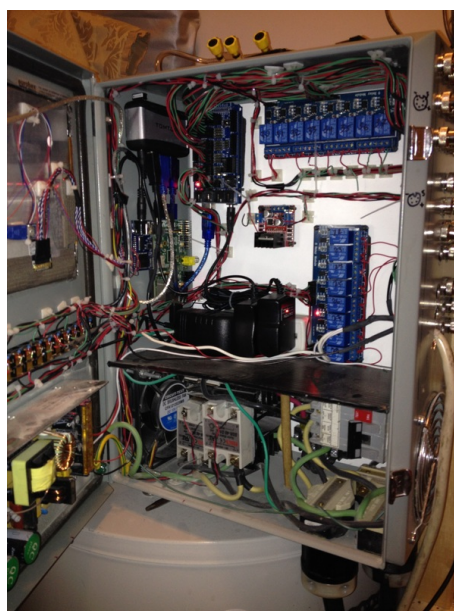
Zařízení	Procesor	RAM	Komunikace
Arduino	ATmega328P	2 kB	I/O porty
BananaPi	ARM Cortex-A7 quad	1 GB	GPIO, USB, WiFi, BT
Raspberry Pi 2B	ARM Cortex-A7 quad	1 GB	GPIO, USB
Raspberry Pi 3B	ARMv8 quad 64bit	1 GB	GPIO, USB, WiFi, BT

## 2.2 Systémy s otevřeným zdrojovým kódem pro vzdálený monitoring

V následující části jsou popsány některé volně dostupné systémy, starající se o monitoring a řízení procesu výroby piva v domácích podmínkách (tzv. domovarnictví – doslovný překlad výrazu Homebrewing podle [3]) a budou porovnány jejich klíčové vlastnosti. Všechny srovnávané jsou postaveny na různých verzích jednodeskového počítače Raspberry Pi.

### 2.2.1 BrewBot

Systém Brew Bot vytvořil strojní inženýr Jason Newton z Manitoby. Srdcem celého systému je tzv. Control Box (na Obr. 2.1) obsahující jak Raspberry Pi, tak i I/O periferie, jako jsou ovládací obvody pump, relé spínače topných těles a monitoring teploty jednotlivých nádob. Pro zprostředkování informací uživateli je na Control Boxu 10" LCD dotykový displej, na kterém je zobrazeno přehledné GUI (Graphical User Interface – Grafické uživatelské rozhraní). Pro vzdálený monitoring je do budoucna připravována webová stránka, umístěná na interním Web serveru. Systém ovšem není modulární (navržen pouze pro práci se dvěma varnými nádobami, kdy jedna slouží jako vystírací, rmutovací a scezovací a druhá pro chmelovar a chlazení). Zdrojové soubory pro tento systém bohužel nejsou momentálně dostupné [9].



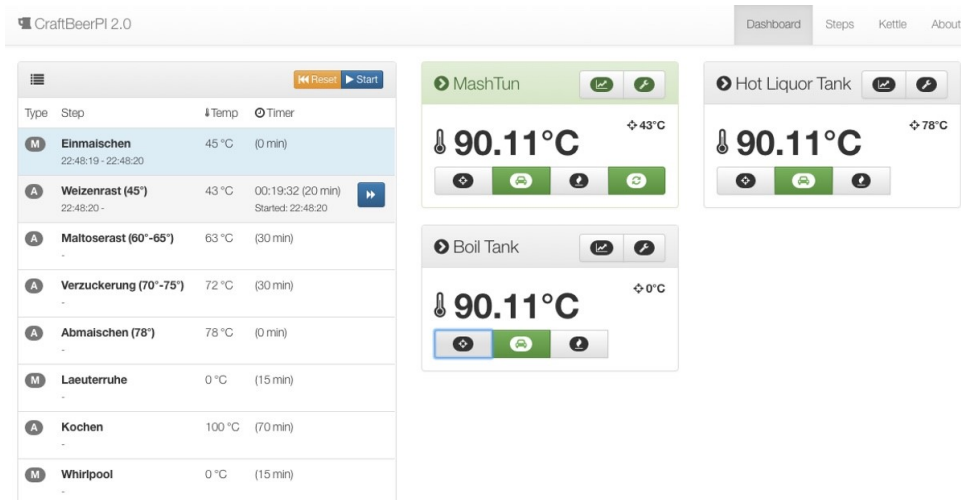
Obr. 2.1: Ovládací skříň systému BrewBot, tzv. Control Box [9]

### 2.2.2 BrewPi

Volně dostupný software BrewPi je velmi pokročilý vůči svým konkurentům. Základním kamenem je zde jednodeskový počítač Arduino Uno, který se stará o řízení a monitorování parametrů a Raspberry Pi, jenž zajišťuje řízení dat, interakci s uživatelem a v neposlední řadě zaznamenávání všech parametrů. Díky tomu lze dosáhnout značné robustnosti systému díky rozdělení rolí. Jednotlivé části totiž mohou fungovat nezávisle. Na integrovaném webovém serveru běží grafické rozhraní, které dává uživateli kompletní přehled nad právě spuštěným programem. Výstup z webu je kompilován a nahrán do Arduina, kde běží samostatně a zpětně předává informace. Program lze za běhu měnit buď z rozhraní, nebo přes integrované ovládání Arduina. K systému je dostupná velmi přehledná dokumentace a návody. Komunita uživatelů je početná. Autoři mají i internetový obchod, na kterém se dají pořídit jak jednotlivé prvky systému, tak i zařízení, nezbytné pro vaření piva v domácích podmínkách [10].

### 2.2.3 CraftBeerPi

Software CraftBeerPi vyniká nad ostatními porovnávanými svojí modulárností a rychlostí úprav systému jeho autorem. Základem je aplikace psaná v programovacím jazyce Python, běžící na počítači Raspberry Pi 2 Model B. Má velmi přehledně zpracované webové GUI (viz Obr. 2.2), které běží na interním web serveru.



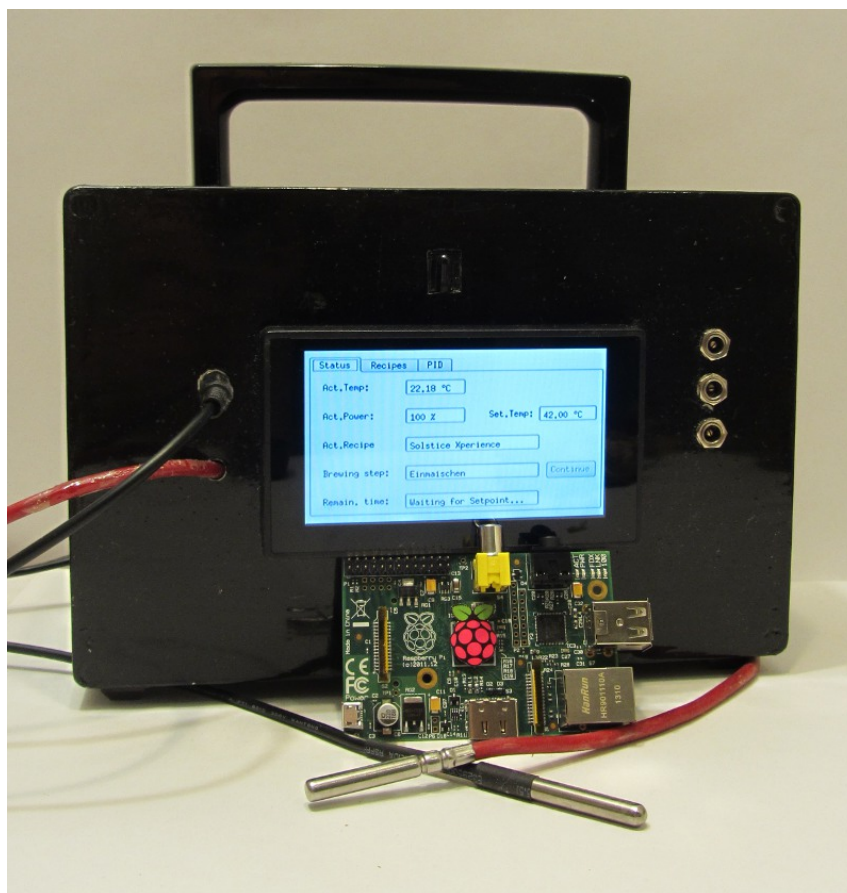
Obr. 2.2: Webová stránka ovládacího rozhraní CraftBeerPi[12]

Konfigurace hardwaru sahají od jedné ovládané varné nádoby s čidlem teploty až po složité celky obsahující až 3 varné nádoby, fermentační nádoby, pumpy a míchadla. Zařízení lze kdykoliv rozšířit díky uživatelsky přívětivému prostředí. Systém

má též nejjednodušší implementaci ze všech. Zdrojové kódy jsou volně dostupné na verzovacím systému GitHub[11], odkud se dají pohodlně nainstalovat pomocí přiloženého skriptu. Komunikace s hardwarem je zajištěna primárně přes GPIO rozhraní [12].

### 2.2.4 MashBerry

MashBerry je jednoduchý systém pro automatizaci procesu vaření piva. Systém je přes GPIO sběrnici spojen s řídicím obvodem, který má na starosti snímání teploty a ovládání periférií, jako jsou topná tělesa, míchadla a pumpy. Pro přehledné zobrazení informací uživateli je k dispozici GUI, zobrazitelné na displeji (viz Obr. 2.3, dá se ovládat i pomocí infračerveného ovladače), nebo přes webovou stránku běžící na interním web serveru. Systém je primárně určen pro vaření v jedné nádobě, ale vzhledem k volně dostupnému zdrojovému kódu se dá relativně jednoduše modifikovat. Softwarově je postavený na upraveném Linuxovém jádře a hlavní části aplikace jsou psány v jazyce C++. Existuje pro něj i velmi přehledná dokumentace, kterou ostatní systémy většinou postrádají [14].



Obr. 2.3: Řídicí jednotka systému MashBerry [13]

### 2.2.5 Vybraný program pro vzdálený monitoring

Jako základ bakalářské práce autor vybral software CraftBeerPi. Toto řešení je dostupné na webových stránkách autora systému, má zdrojový kód psaný v jazyce Python, dostupný z GitHubu[11]. Jeho implementace je velmi snadná. Stačí nainstalovat nejzákladnější balíčky, zkopírovat zdrojové soubory a spustit instalační skript. Postup instalace bude uveden ve 4. kapitole této práce. Schémata se základním zapojením hardwaru jsou dostupná na internetu díky široké komunitě uživatelů. Nejjednodušší zapojení je uvedeno i v dokumentaci na GitHubu. Srovnání s ostatními je v Tab. 2.4.

Tab. 2.4: Tabulka srovnání zvolených programů pro řízení procesu výroby piva

Název	Programovací jazyk	Web server	Zařízení
BrewBot	Python, PHP	lighttpd	RPi/Arduino
BrewPi	Python, PHP	LAMP web server	RPi2B/Arduino
Mashberry	C++, PHP	LAMP web server	RPi1B
CraftBeerPi	Python, Javascript	Flask	RPi3B

## 3 NÁVRHY AUTOMATIZOVANÉ VARNY

Tato kapitola se bude zabývat výběrem vhodných komponent domácí automatizované varny. Hlavním cílem je popsat jednotlivé možnosti základního uspořádání, prvky použitelné pro vzdálený monitoring a prvky pro automatizované řízení procesu výroby piva. V praxi domácích minipivovarů se setkáme nejčastěji s použitím Infúzního rmutování a mnoha způsoby uspořádání varného systému (např. RIMS nebo HERMS). Pro tuto práci byl zvolen návrh systému v uspořádání RIMS neboť je jeho konstrukce relativně jednoduchá na realizaci při zachování velmi dobré funkčnosti celku.

### 3.1 Varné nádoby

Varná nádoba je základním stavebním kamenem celého systému. Odehrávají se v ní všechny procesy související s výrobou piva, je tudíž nezbytné aby byla vyrobená z materiálů, které jsou zdravotně nezávadné a mají dostatečnou odolnost i při teplotách kolem 100 °C. Musí být umístěna na regulovatelném zdroji tepla, nebo takovýto zdroj přímo obsahovat (např. ve formě topného tělesa). Regulace je provedena buď manuálně, kdy je celý systém kontrolován a podle potřeby regulován tepelný výkon, nebo automaticky pomocí termostatu, v případě této bakalářské práce obvody řídicí elektroniky. Pro lepší manipulaci s obsahem je dobré do spodní strany takového nádoby umístit vypouštěcí ventil.

#### 3.1.1 Úprava pro vystírání

Varná nádoba určená pro vystírání musí obsahovat buď na dně umístěnou děrovanou desku, která zabrání pevným zbytkům zrn proniknout k topným tělesům a při závěrečné fázi vystírání poslouží ke scezení sladiny, nebo musí být filtrace zajištěna jinak. Topná tělesa nemusí mít vysoký výkon, protože teplota je v počátečních fázích zvedána velmi pozvolna. Nádoba však musí umožňovat regulaci teploty a její zpětný odečet (například pomocí elektronického nebo mechanického teploměru). Lze použít i zařízení pro sledování objemu nádoby. Při vystírání je potřeba obsah stále promíchávat. Pro případ automatizace je tedy vhodné, aby vystírací nádoba obsahovala elektromotor, pohánějící míchadlo, nebo vhodný recirkulační systém. Jinak by v této fázi musela být přítomna obsluha, jenž by zajišťovala míchání. Při Infúzním rmutování by docházelo k postupnému zvyšování teploty obsahu až na bod varu. Nádoba by pro takovouto situaci musela být vhodně uzpůsobena (vyšší výkon topného tělesa, materiály odolné vůči vyšším teplotám).





Obr. 3.1: Varná souprava v konfiguraci HERMS

### 3.1.2 Chmelovar

V případě konečného vaření a chmelení nejsou na nádobu kladeny takové požadavky jako při vystírání. Nádoba by měla obsahovat síto, pro oddělení pevných částic po ukončení vaření. Musí umožnit rychlé dosažení bodu varu, vhodnou regulaci a zpětnou vazbu teploty. Při Dekokčním rmutování může být část sladiny převedena do chmelovaru, kde může projít bodem varu. Díky tomu odpadne potřeba samostatné rmutovací nádoby. Budoucí mladinu není potřeba tak často míchat, ale použití automatického míchadla je opět výhodné. Po dokončení vaření musí být mladina zchlazena, což může být provedeno použitím vhodného chladicího systému.

### 3.1.3 Nádoba na horkou vodu

Některé typy uspořádání (např. HERMS) využívají samostatnou nádobu na ohřev vody, která bude dále zužitkována v procesu vystírání a rmutování. Touto nádobou prochází spirála z měděných trubek, která slouží k recirkulaci sladiny a jejímu postupnému ohřevu. Díky tomu odpadá potřeba ohřívání vystírací nádoby. Po doplnění objemu odpařené vody nedochází ke schlazení várky.

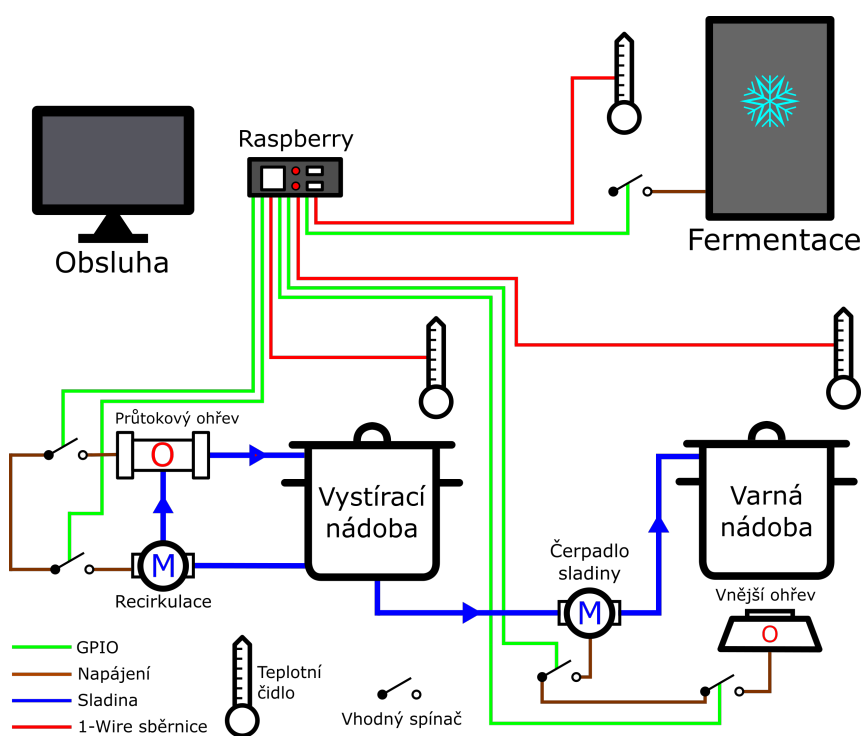
### 3.1.4 Čerpadla

Čerpadla usnadňují převod kapalin mezi jednotlivými nádobami (např. při rmutování a scezování). Pro automatizaci výroby není přímo potřebné, nicméně usnadňuje

práci a zrychluje celý proces výroby. Vhodné čerpadlo musí mít dobrou teplotní odolnost, musí zvládnout přečerpání i menší částice obsažené ve várce a musí být certifikováno pro použití v potravinářství. V opačném případě riskujeme jeho poškození, nebo kontaminaci dávky například mazací směsí.

## 3.2 Elektronika systému s uspořádáním RIMS

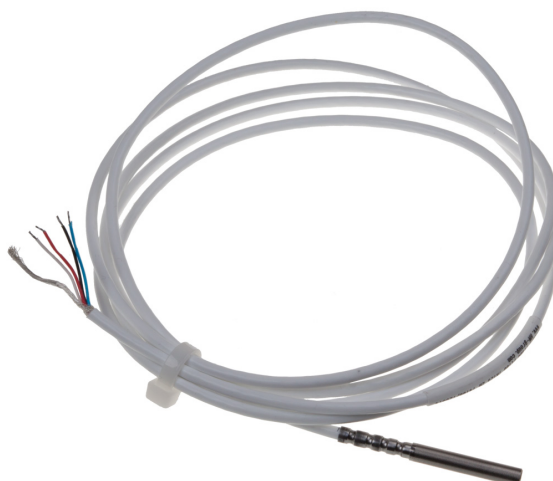
Uspořádání varné soupravy v systému RIMS se nejčastěji skládá ze dvou varných nádob, propojených mezi sebou systémem trubek, ventilů a čerpadel a z nádoby fermentační. V jedné nádobě dochází k vystírání a infúznímu rmutování a vzniká v ní sladina. Poté je celá várka přečerpána do druhé nádoby, kde dochází ke vzniku mladiny. Ve fermentační nádobě dochází k samotnému kvašení a dozrávání.



Obr. 3.2: Znázorněný příklad vzorového uspořádání systému typu RIMS

### 3.2.1 Řídící jednotka

Hlavním prvkem řídicí jednotky je jednodeskový počítač z rodiny Raspberry. Jednotlivé modely jsou voleny podle potřeb a cenových možností uživatele (v této práci bylo zvoleno Raspberry Pi model 3 B). Ovládání systému je zajištěno obsluhou z připojeného počítače, tabletu nebo jiného zařízení, umožňujícího připojení na webovou



Obr. 3.3: Vodotěsné teplotní čidlo DS18B20 [17]

stránku aplikace CraftBeerPi. Konektivita s obsluhujícím zařízením je tedy zajištěna připojením na lokální síť a to jakýmkoliv podporovaným způsobem. Na aplikaci CraftBeerPi lze přistoupit i přímo ze zařízení, pokud má Raspberry připojené vhodné periferie (klávesnice, myš a monitor). Připojení hardwaru je provedeno kabeláží propojenou s porty GPIO.

### 3.2.2 Měření teploty

Měření teploty v jednotlivých nádobách má jednak funkci informativní, poskytuje provozovateli aktuální data a dá se tak sledovat vývoj teploty v průběhu cyklu, hlavně ale slouží jako zpětná vazba řídicím obvodům pro přesné nastavení teploty spínacím obvodem. Pro automatizaci procesu výroby piva je potřeba aby teploměr měřil v rozsahu 0-120 °C. Dalším důležitým požadavkem je vodotěsnost, neboť teplota je odečítána přímo z kapaliny. Existuje mnoho různých řešení pro odečtení teploty: jednodrátové teploměry (1-Wire interface firmy Dallas), moduly, komunikující přes I2C sběrnici, teploměry s analogovým výstupem a mnoho dalších. V systému CraftBeerPi je použito jednodrátové teplotní čidlo DS18B20 (viz 3.3, komunikující přes vhodný komunikační obvod na rozhraní GPIO04 [12]. Další alternativou je platinové teplotní čidlo PT100 a mnohá další. Cena těchto senzorů se pohybuje od cca 50 do několika set Kč, v závislosti na provedení [17].

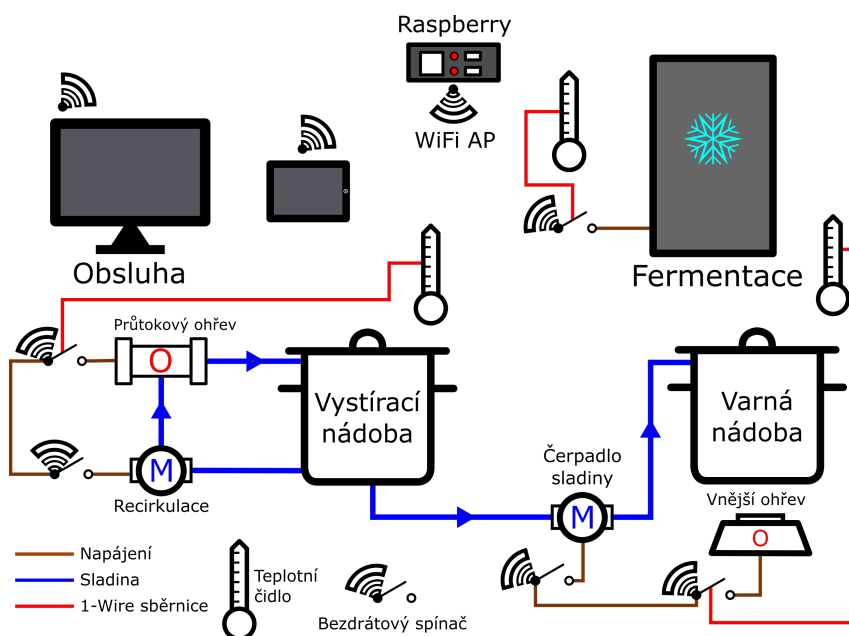
### 3.2.3 Výkonové spínání

Výkonová část musí být schopná spínat relativně velké proudy neboť příkon topných těles ve varné části se může lehce překročit 3kW. Pro spínání velkých výkonů se

používají převážně relé, výkonové unipolární tranzistory, tyristory a v neposlední řadě Solid state relé (SSR). Autor pro vzorový návrh zvolil právě SSR, neboť je použito ve zvolené aplikaci pro vzdálený monitoring [12]. SSR používají i ostatní softwarová řešení, probíraná v předchozích kapitolách (viz 2.2). Velkou výhodou je i příznivá pořizovací cena těchto polovodičových součástek [18].

### 3.3 Elektronika systému s uspořádáním RIMS rozšířený o bezdrátovou komunikaci

Uspořádání systému s bezdrátovou komunikací je stejné jako v části 3.2. Hlavní rozdíl je v tom, že jednotlivé moduly mezi sebou nekomunikují pomocí sběrnice GPIO, ale pomocí vhodné bezdrátové technologie. Velmi jednoduchým řešením je použití standardu WiFi nebo Bluetooth. WiFi nabízí vyšší přenosové rychlosti a lepší přístup signálu překážkami. Použití bezdrátové lokální sítě je navíc velmi jednoduché. Bluetooth má velkou výhodu díky nízké energetické náročnosti. Vzhledem ke spínání síťového napětí se však tato stává nepodstatnou. Na trhu existuje mnoho zařízení, které mezi sebou dokáží komunikovat pomocí bezdrátové lokální sítě na frekvenci 2,4 GHz. Pro aplikaci v systému pro vzdálený monitoring lze využít bezdrátové WiFi spínače, teploměry a mnohé další.



Obr. 3.4: Znáznornění systému RIMS, který komunikuje bezdrátově

### 3.3.1 Řídící jednotka

Jako řídicí jednotka opět slouží jednodeskový počítač Raspberry. Tentokrát je ovšem nejvýhodnější použít Model 3 B, jenž má zabudovaný WiFi modul. Lze využít i jiné modely s externě připojeným WiFi zařízením. Jednotka, nastavená jako bezdrátový přístupový bod připojí všechna ostatní zařízení do své vlastní lokální sítě, odkud se budou ovládat. Do stejné sítě lze připojit i zařízení jako tablet nebo počítač, které můžou sloužit pro připojení k webovému rozhraní CraftBeerPi.

### 3.3.2 Bezdrátové spínání

Výkonové spínače musí být pro příjem povelů z řídicí jednotky dovybaveny vhodným bezdrátovým modulem. Tento modul zprostředkuje předání informací a sepnutí zařízení na výstupu. Jako spínač může být použito opět relé, solid state relé, unipolární tranzistor či tyristor. Povel k sepnutí ovšem nevysílá Raspberry přes GPIO, nýbrž zprostředkovávající zařízení (např. ESP8266).

Na trhu existuje takových zařízení mnoho. Například bezdrátové spínače Sonoff. Jedná se o WiFi IoT spínač firmy Itead. Je tvořen napájecím modulem, bezdrátovým mikrokontrolerem ESP8266 a výstupním relé. Zařízení je primárně určeno k propojení s chytrým telefonem nebo tabletem s Android OS. Existuje ovšem řada článků, které se zabývají možností přeprogramování originálního řídicího programu a ovládání přes Raspberry [19].



Obr. 3.5: Bezdrátový spínač Sonoff [19]

### 3.3.3 Bezdrátová komunikace s měřicím čidlem

I v problematice bezdrátové komunikace s čidly teploty existuje mnoho možností, jak teplotu odečíst a bezdrátově předat řídicí jednotce. Autor se rozhodl v této bakalářské práci použít teplotní senzor DS18B20. Pro vytvoření zprostředkovávající 1-Wire sběrnice opět slouží modul ESP8266, který disponuje I/O porty, které lze



Obr. 3.6: Bezdrátový spínač Sonoff TH [20]

použít pro její realizaci. Díky dostupnosti několika portů lze zařízení ještě zkombinovat se spínacím obvodem. Takto utvořený celek lze využít ke kompletnímu ovládání varné nádoby, neboť je zajištěno jak spínání, tak i zpětná vazba teploty. Firma Itead takovéto zařízení nabízí. Jedná se o IoT spínač Sonoff TH, který v sobě kombinuje 10 nebo 16 ampérové relé a vstup pro zařízení na 1-Wire sběrnici [20]. Oba dva zmíněné IoT spínače jsou součástí produktové řady Sonoff této společnosti. Základní vlastnosti některých zařízení této řady jsou uvedeny v Tab. 3.1.

Tab. 3.1: Tabulka vlastností spínačů Sonoff

Název	WiFi	Napětí	Maximální proud	Přídavné funkce
Sonoff	802.11 b/g/n	90-250 V	10 A	Žádná
Sonoff RF	802.11 b/g/n	90-250 V	10 A	433 MHz ovládání
Sonoff TH10/16	802.11 b/g/n	90-250 V	10/16 A	1-Wire Interface
Sonoff 4ch	802.11 b/g/n	90-250 V	10 A	4-kanálový spínač
Sonoff Pow	802.11 b/g/n	90-250 V	16 A	Měření proudu

## 4 IMPLEMENTACE SOFTWARE PRO VZDÁLENÝ MONITORING

Vybraný software pro vzdálený monitoring a řízení procesu výroby piva bylo třeba nainstalovat na jednodeskový počítač Raspberry Pi. Vzhledem ke kompatibilitě byl zvolen operační systém Raspbian od Raspberry Pi Foundation, na kterém jsou jednotlivé balíčky odzkoušeny a jenž je distribuován jako hlavní operační systém pro Raspberry Pi [21]. Software by vhodnou úpravou instalačního skriptu měl jít instalovat i na ostatní Linuxové operační systémy nicméně autoři CraftBeerPi doporučují právě Raspbian [11].

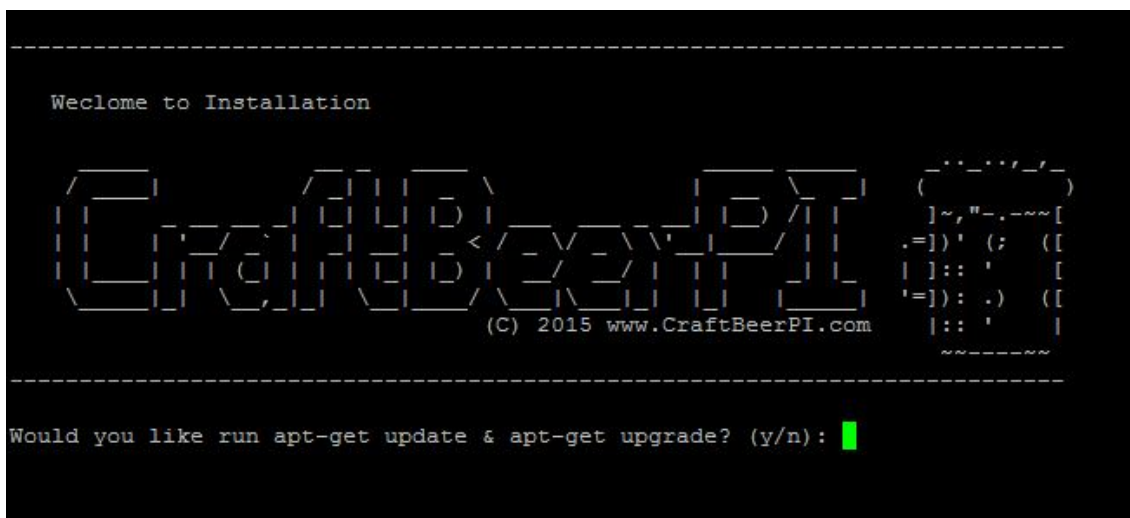
### 4.1 Instalace operačního systému Raspbian

Instalace operačního systému Raspbian spočívá ve stažení obrazu disku ze webu distributora [21]. Tento obraz je třeba umístit na vhodné paměťové médium (v případě Raspberry Pi 3 se jedná o MicroSD kartu o velikosti minimálně 4 GB). To lze provést několika způsoby. Autor se rozhodl postupovat podle základního popsaného postupu, který spočívá v kopírování souborů pomocí softwaru Win32DiskImager [22]. Po dokončení kopírování je paměťová karta umístěna do jednodeskového počítače, jsou připojeny vhodné periferie (klávesnice, monitor nebo displej a připojení k lokální síti) a po připojení napájení dochází k zavedení systému. Při prvotním spuštění slouží jako přihlašovací údaje uživatelský účet `pi` s heslem `raspberrypi`. Toto výchozí nastavení je vhodné z důvodu větší bezpečnosti upravit a to minimálně změnou hesla. Dále je vhodné použít statické nastavení IP adresy a aktualizovat systém. Pro vytváření GUI rozhraní autor nainstaloval X Windows system, grafickou nástavbu Pixel a pro připojení vzdálené plochy TightVNC server. Při přechodu na statickou IP adresu se vyskytl problém s DHCP klientem, který se snažil síťové nastavení získat z DHCP serveru a ignoroval statické nastavení, takže byly na zařízení 2 IP adresy ve stejnou chvíli. Tento problém byl vyřešen odstraněním balíčku s DHCP klientem. V této fázi je systém připraven na instalaci souborů CraftBeerPi.

### 4.2 Instalace CraftBeerPi

Všechny instalované zdrojové soubory jsou umístěny na GitHubu jako software licencí GNU GPL. Dají se tedy volně stáhnout, prohlížet a dle potřeby upravovat. Je zde popsán i jednoduchý návod na instalaci. Příkazy se spouštějí s právy superuživatele [11].





Obr. 4.1: Úvodní obrazovka instalačního skriptu

- Prvním krokem je propojení verzovacího systému GitHub s Raspbianem a instalace jádra GIT. To lze provést příkazem `apt-get install git-core`. Správce balíčků nainstaluje automaticky všechny potřebné soubory a zavede je do systému.
- Příkazem `git clone https://github.com/Manuel83/craftbeerpi.git` jsou staženy soubory CraftBeerPi.
- Nyní je potřeba přepnout se do stažené složky `craftbeerpi` a spustit instalační skript `install.sh`.

Skript automaticky stahuje závislosti, potřebné pro funkci systému. Uživatel je nejprve vyzván, zda se má provést update a upgrade balíčků operačního systému, poté začíná samotná instalace. Instaluje se například rozhraní `wiringPi`, nezbytné pro komunikaci přes GPIO, implementace programovacího jazyka Python včetně mikroframeworku `Flask` a mnohé další. Všechny potřebné závislosti jsou vypsány v souboru `requirements.txt`. Poslední fází je instalace podpory pro 1-Wire sběrnici, sloužící pro připojení teplotních čidel, spuštění po startu operačního systému a nakonec je Raspberry restartováno. Zdrojové soubory zůstávají ve složce, kam bylo provedeno klonování a odtud se i provádí spouštění serveru (příkaz `python runserver.py`).

## 4.3 Webová aplikace

Webová aplikace je hlavním řídicím prvkem softwaru CraftBeerPi. Serverová část (Obr. 4.2 je tvořena mikroframeworkem `Flask`, jenž pomocí definovaných Python skriptů umožňuje ovládání hardwaru, správu databáze, a komunikaci mezi API



```

None
192.168.0.1 - - [2016-11-17 20:49:39] "GET /api/kettle/devices HTTP/1.1" 200 369 0.004103
192.168.0.1 - - [2016-11-17 20:49:39] "GET /api/hardware HTTP/1.1" 200 399 0.024640
192.168.0.1 - - [2016-11-17 20:49:40] "GET /api/config HTTP/1.1" 200 1606 0.038219
192.168.0.1 - - [2016-11-17 20:49:41] "GET /api/kettle HTTP/1.1" 200 395 0.040557
192.168.0.1 - - [2016-11-17 20:49:41] "GET /api/kettle/devices HTTP/1.1" 200 369 0.004943
192.168.0.1 - - [2016-11-17 20:49:41] "GET /api/automatic/paramter HTTP/1.1" 200 438 0.004
192.168.0.1 - - [2016-11-17 20:49:41] "GET /api/kettle/thermometer HTTP/1.1" 200 117 0.005
192.168.0.1 - - [2016-11-17 20:49:42] "GET /api/step HTTP/1.1" 200 1395 0.041577
192.168.0.1 - - [2016-11-17 20:49:42] "GET /api/config HTTP/1.1" 200 1606 0.040622
192.168.0.1 - - [2016-11-17 20:49:42] "GET /api/kettle HTTP/1.1" 200 395 0.027172
192.168.0.1 - - [2016-11-17 20:49:48] "POST /api/step/clear HTTP/1.1" 204 123 0.059537
192.168.0.1 - - [2016-11-17 20:49:49] "GET /api/step HTTP/1.1" 200 395 0.040407
192.168.0.1 - - [2016-11-17 20:49:49] "GET /base/static/partials/steps/recipe_book_overvie
1107 0.011184
192.168.0.1 - - [2016-11-17 20:49:50] "GET /api/recipe_books HTTP/1.1" 200 407 0.034514
192.168.0.1 - - [2016-11-17 20:49:55] "GET /api/recipe_books HTTP/1.1" 200 407 0.024213
192.168.0.1 - - [2016-11-17 20:49:56] "GET /base/static/partials/steps/save_recipe.html HT
0895
192.168.0.1 - - [2016-11-17 20:49:59] "GET /base/static/partials/steps/form.html HTTP/1.1"
192.168.0.1 - - [2016-11-17 20:50:07] "GET /api/recipe_books HTTP/1.1" 200 407 0.032615

```

Obr. 4.2: Server, běžící v příkazové řádce

CraftBeerPi a HTML a JavaScriptovou částí webového rozhraní. Komunikace mezi klientem a serverem probíhá v reálném čase pomocí obousměrného komunikačního kanálu `socket.IO`.

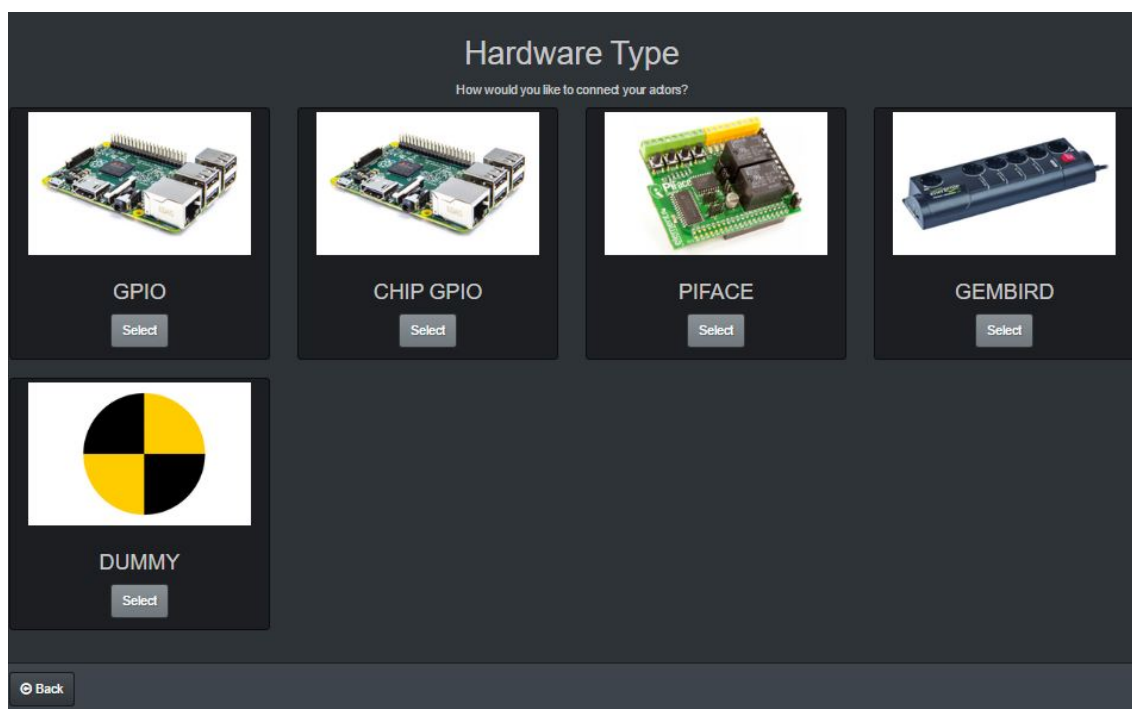
### 4.3.1 První spuštění

Po spuštění souboru `runserver.py` se lze připojit na webové rozhraní CraftBeerPi. To je dostupné na IP adrese Raspberry přes port 5000. Připojení není nijak zabezpečeno. Nejprve je uživatel vyzván k prvotnímu nastavení, poté již může celý systém ovládat. V nastavení je nejprve zvolen výstup ovládání. Na výběr máme rozhraní GPIO, PiFace (HW ovladač od autorů CraftBeerPi), chytrou zásuvku Gembird a nakonec zařízení DUMMY, jenž slouží k testovacím účelům a simuluje fungující hardware (viz Obr. 4.3).

Dalším bodem nastavení je výběr teploměru. Zde máme možnost připojit 1-Wire zařízení na portu GPIO04. Logická adresa připojených senzorů se načítá automaticky. Dále je zde k dispozici teploměr pro testovací účely DUMMY. Ten je naprogramován s předem nastavenou teplotou a lze díky němu ověřit funkčnost spínání.

Poté dochází k nastavení spínaných zařízení a adresaci všech teplotních čidel. Ke každému zařízení lze přiřadit jméno, korespondující port a v neposlední řadě jednu z použitých přednastavených logik, které budou sloužit k ovládání daného hardwaru.

Varné nádoby jsou nastaveny nakonec. Zde dojde k přiřazení GPIO výstupu spínání, míchání a k přiřazení teploměru. Lze nastavit i název varné nádoby a její rozměry.



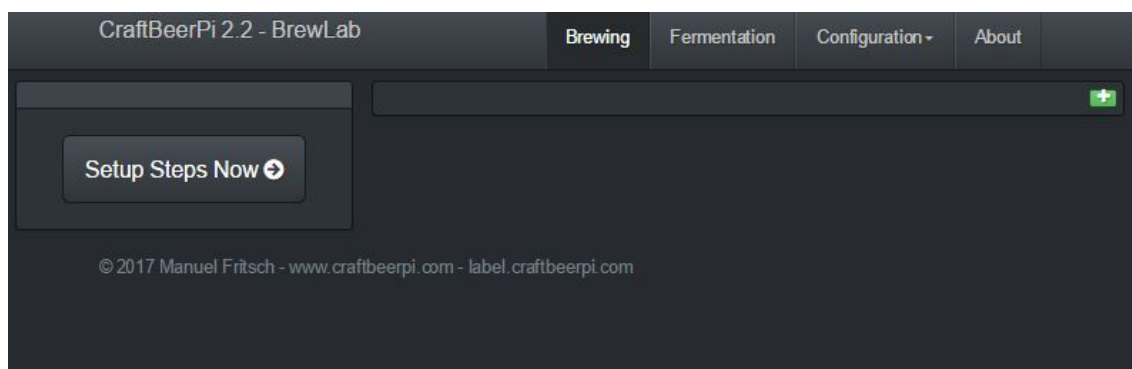
Obr. 4.3: Nastavení systému

### 4.3.2 Úvodní obrazovka

Na úvodní obrazovce (na Obr. 4.4) můžeme vidět přehled aktuální činnosti softwaru. Na levé straně se zobrazují jednotlivé kroky vaření. Každý krok má název, teplotu, které bude dosaženo a dobu trvání. Je zde umístěna i informace o typu kroku (M pro manuálně ovládaný, A pro automatický). Pravá strana zobrazuje informace o připojeném hardwaru a nastavených nádobách v podobě ovládacích ploch. Horní strana obsahuje název a tlačítko pro vstup do nastavení nádoby, uprostřed je zobrazena aktuální teplota (po otevření se zobrazí historie teploty v přehledném grafu). Ve spodní straně jsou umístěny ovládací prvky pro nastavení teploty, přepínání automatického režimu, spínání ohřevu a míchadla/recirkulace. Pro každou varnou nádobu a připojený hardware je vykresleno jedno ovládací pole, jenž je průběžně aktualizováno. Zobrazení spínacích tlačítek hardwaru a zobrazovací pole teploty lze při jejich nastavování skrýt.

### 4.3.3 Fermentace

Položka menu **Fermentation** ukrývá ovládací rozhraní pro správu fermentačních nádob. Každá fermentační nádoba umožňuje připojit až dvě ovládaná zařízení a teplotní čidlo. Protože pivovarské kvasinky jsou velmi citlivé na teplotu, používá se u fermentační nádoby jak chlazení tak i spínání ohřevu. Naopak promíchávání ob-



Obr. 4.4: Úvodní stránka CraftBeerPi

sahu fermentační nádoby již není potřeba. Ve správě kroků fermentace lze spravovat položky obsahující teplotu a čas, po který má být držena (v řádu dnů). Lze nastavit i logiku spínání (dosažení teploty přesně, s hysterezí atd.).

#### 4.3.4 Správa kroků a receptů

Na úvodní obrazovce se též nachází správce jednotlivých kroků vaření a receptů. Ovládací pole obsahuje tlačítka pro přidávání a odebírání kroků, zobrazení seznamu receptů a import z programu Kleiner Brauhelfer.<sup>1</sup> Dále můžeme vidět chronologicky seřazené informace o jednotlivých úkolech. Pro každou položku seznamu je možná její úprava (logika, teplota, časovač a varná nádoba) a ukládání do knihy receptů.

#### 4.3.5 Ostatní funkce

Pod položkou menu **Hardware** nalezneme nástroj pro pohodlnou správu varných nádob a připojeného hardwaru. Zde můžeme přidávat, odebírat a upravovat parametry a přidávat nově připojená zařízení. Editační menu je zobrazeno velmi přehledně a ovládání je intuitivní. Každá nádoba obsahuje doplňující informace jako jsou rozměry, logika nádoby (PID, hystereze, atd.), nejdůležitější údaje však jsou výstupy spínání hardwaru a název teploměru. Každá jednotlivá položka připojených zařízení zde může být editována (název, výstup GPIO) jak je vidět na Obr. 4.5.

Následuje položka **Configuration**. Zde se dají nastavit základní parametry serveru. Předně se jedná o nastavení adresace 1-Wire interface pro teploměry, možnosti připojení hardwaru a mnohé další. Po každé změně hardwaru je nezbytné server restartovat. To lze provést v menu **About** odpovídajícím tlačítkem, nebo z příkazové řádky.

<sup>1</sup>jedná se o volně dostupný software pro vytváření receptur doma vařených piv

Kettle	Thermometer	Heater	Agitator	Diameter	Height	Delete
Kettle1	--PLS_! ▾	--PLS_ ▾	--PLS_ ▾	<input type="text"/>	<input type="text"/>	
Kettle2	--PLS_! ▾	--PLS_ ▾	--PLS_ ▾	<input type="text"/>	<input type="text"/>	
Kettle3	--PLS_! ▾	--PLS_ ▾	--PLS_ ▾	<input type="text"/>	<input type="text"/>	

Back Next

Obr. 4.5: Nastavení varné nádoby

## 4.4 Úpravy aplikace CraftBeerPi pro řízení bezdrátových spínačů

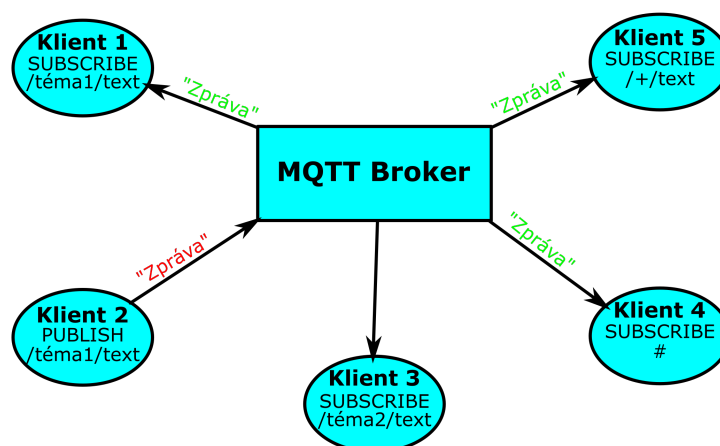
Pro spolehlivé řízení spínačů (aktorů) a bezdrátové měření teploty bylo nutné původní aplikaci CraftBeerPi vhodně upravit. Funkce bezdrátového ovládání a připojení zařízení v programu CraftBeerPi částečně existuje. V nastavení připojení hardwaru lze jako **Actor type** zvolit bezdrátovou zásuvku **WiFi socket**. Její nastavení však není dotaženo do konce a dokumentace k softwaru se o této možnosti zmiňuje jen okrajově. Pro připojení bezdrátových teplotních čidel neexistuje možnost vůbec žádná. Podporu bezdrátových zařízení tedy bylo potřeba vytvořit.

Vzhledem ke složitosti celého softwarového řešení CraftBeerPi bylo nejprve potřeba zvolit vhodnou formu bezdrátového přenosu dat a vhodný protokol, který by tento přenos řídil. Možnost použití Bluetooth byla zavržena již v začátku, protože práce s bezdrátovou lokální sítí je jednodušší a uživatelsky příjemnější. Jako komunikační protokol byl nejprve zvolen HTTP. Tím byla jednak ověřena funkčnost spínání a také možnost úpravy samotného skriptového pozadí webové aplikace. Nicméně pro těžkopádnost protokolu a problémy se zpětným přenosem dat byl využit protokol MQTT.

#### 4.4.1 Protokol MQTT a jeho implementace do CraftBeerPi

Message Queuing Telemetry Transport vznikl jako nenáročný protokol pro komunikaci mezi IoT zařízeními kdy nahradil nevyhovující protokol HTTP. Oproti němu využívá mnohem jednodušší konstrukce jednotlivých zpráv a povelů. Funguje na principu subscriber-publisher, kdy se všechny zprávy přeposílají přes centrální bod (Broker), který se stará o jejich správné rozesílání a správu témat (Topic) a jejich úrovní (úrovně jsou odděleny znakem /), do kterých jsou jednotlivé zprávy tříděny. Broker tyto zprávy přeposílá zařízením, které se přihlásily k jejich odběru (Subscribe). Zprávu, která nemá žádný předem daný formát, jakékoliv zařízení k danému tématu publikuje (Publish). Velikost může být až 256 MB ale zpravidla jsou posílány mnohem menší objemy dat. Takto velkou zprávu navíc často nepodporuje aplikační rozhraní klientů. Minimalizace MQTT je docílena tak, že je přidáváno jen nezbytně nutné minimum servisních informací. Protokol zavádí tři úrovně Quality of Service – kvalita služeb (QoS) [23].

Jako Broker v této bakalářské práci figuruje program Eclipse Mosquitto s otevřeným zdrojovým kódem. Jedná se o odlehčenou MQTT 3.1 serverovou aplikaci, umožňující připojení všem aktivním MQTT klientům. V rámci nastavení lze provést jednoduché (uživatelské účty), ale i pokročilé (SSL, TLS) možnosti zabezpečení [24]. Pro tuto práci nebylo zabezpečení serveru použito, neboť bude naslouchat pouze na zabezpečené bezdrátové síti, na kterou se všechny IoT spínače budou připojovat.



Obr. 4.6: Příklad funkce protokolu MQTT

Jednotlivá připojená zařízení musí být nastavena k publikování požadovaných parametrů v periodických časových intervalech. Každé z nich bude vysílat zprávu s tématem, obsahujícím název zařízení na první úrovni a název přenášené zprávy ve druhé (/NazevSpinace/TemaZpravy). Díky tomu bude vždy vytvořeno samo-

statné téma pro jednotlivá připojená zařízení. Informace o teplotě (Temp) a relativní vzdušné vlhkosti (Humidity – pouze pokud ji snímač podporuje) se bude posílat každou vteřinu od připojení k Brokeru. Pro testovací účely bude vysílána ještě informace o úrovni přijímaného signálu bezdrátové sítě a to každých 10 sekund. IP adresa Brokera je 10.0.1.2 se standardním MQTT portem je port 1883. Povel pro sepnutí výstupu bude publikován zprávou k tématu obsahujícím opět název zařízení na první úrovni a příznak k vykonání příkazu na druhé. (/NazevSpinace/cmd)

Na straně aplikace CraftBeerPi dochází ke čtení a publikování MQTT zpráv pomocí knihovny MQTT klienta Eclipse Paho MQTT. Tato knihovna umožňuje integraci MQTT klienta přímo v programovacím jazyce Python ve kterém je pána celá serverová aplikace. Zavádí tři nové třídy pro práci se zprávami. Třída Client vytváří instanci klienta pro kontinuální přijímání a publikování zpráv. Třída Publish vytváří instanci pro jednoduché a rychlé odesílání MQTT zprávy ke zvolenému tématu. Je volána pouze pokud je potřeba odeslat informace Brokeru a narozdíl od třídy Client nemusí nejprve vytvořit spojení se serverem. Pro získání jedné MQTT zprávy k přihlášenému tématu slouží třída Subscribe. Ke knihovně je dostupná rozsáhlá dokumentace, která popisuje práci s jednotlivými třídami a funkcemi [25].

#### 4.4.2 Vytvoření seznamu MQTT klientů

O vytvoření seznamu připojených klientů se stará soubor `mqttHandler.py` a je k němu využita třída `paho.mqtt.client`, která se stará o navázání spojení a přihlášení k odběru témat MQTT Brokera. Program pracuje následujícím způsobem: Nejprve je vytvořena instance třídy `mqtt.Client()`. Poté dochází k nastavení IP adresy Brokera a připojení klienta `client.connect()`. To vyvolá událost `client.on_connect` a volá se funkce pro nastavení odebíraného tématu. Vzhledem k tomu, že každé připojené zařízení bude vysílat zprávy pod svým vlastním tématem, které obsahuje i název zařízení, je nutné aby klient odebíral všechna dostupná témata Brokeru a získal tak informaci o jméně od každého připojeného zařízení. Pro tento účel lze využít zástupné znaky, kdy `+` zastupuje jakoukoliv jednu úroveň subskribce a `#` všechny následující (v tomto případě tedy znak `#` na první úrovni znamená odběr všech témat). Po úspěšném přihlášení k odběru začnou klientovi přicházet jednotlivé zprávy, což vyvolá událost `client.on_message`. Ta volá funkci `onMessage` a předává jí zprávu `msg`. Z jejího tématu (`msg.topic`) je poté pomocí práce s řetězci a seznamy vyselektován název zařízení a uložen do globální proměnné `names`. Funkce `getNames` vrací programům, které ji zavolají seznam jmen připojených zařízení. Klient, běžící v nekonečné smyčce v samostatném vlákně programu, zprávy načítá neustále až do ukončení aplikace, aby došlo zaznamenání jmen všech zařízení, připojených k brokerovi. Soubor je umístěn ve složce `/brewap/base` a je k nahlédnutí v příloze A.1

[26].

### 4.4.3 Řízení vstupně/výstupních portů protokolem MQTT

Pro přijetí povelu každé z připojených zařízení odebírá téma `/NazevZarizeni/cmd`. K sepnutí výstupu je jako MQTT zpráva poslán text `'GPIO,12,1'`. GPIO označuje příkaz který se má vykonat, první číslo označuje konkrétní GPIO výstup a poslední číslo označuje logickou hodnotu, která bude danému výstupu přiřazena. O správnou funkci bezdrátového spínání se stará autorem vytvořený skript `espeasy.py` (k nahlédnutí v příloze A.2). Nachází se ve složce s Python frameworkem CraftBeerPi (složka `/brewap/base/devices`) pro manipulaci s hardwarem a částečně vychází z původního souboru pro práci s GPIO výstupy `gpio.py`. Původní zdrojový soubor byl však změněn na práci s MQTT protokolem. Jako název zařízení slouží přímo název aktoru, získaný z databáze. O správné provázání databáze a webové aplikace, funkci spínání a logiku se starají původní programy `actor.py`, `config.py`, `model.py`, `setup.py` a `system.py`. Tyto soubory byly upraveny pouze přidáním odkazů na soubory nové [26].

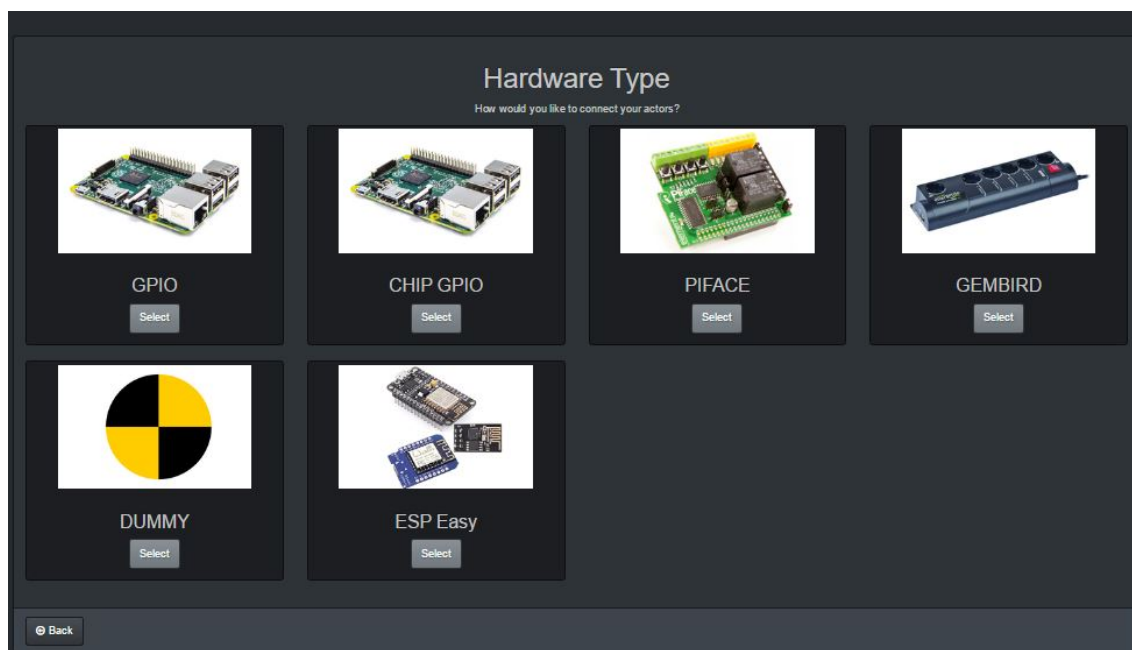
### 4.4.4 Získání teploty z MQTT zprávy

Soubor pro získání teploty ze senzoru se nachází ve složce `/brewap/base/thermometer` spolu s ostatními programy pro čtení teploty z různých zařízení. Nachází se zde soubory `dummy_thermometer.py`, `usb_thermometer.py` a `w1_thermometer.py`, které mají za úkol číst teplotu na předem zvolených sběrnicích. Vytvořený soubor `esp_thermometer.py` (viz příloha A.3) možnosti aplikace rozšiřuje o načítání informací o teplotě z MQTT zprávy. Pro přenos zprávy od Brokera slouží vytvořený jednorázový kanál `subscribe.simple()`, kterému jsou předány informace o spojení a jméno senzoru. Od tohoto senzoru je přijata právě jedna zpráva s tématem `/NázevSenzoru/temp`. Zpráva obsahuje přesnou informaci o aktuální teplotě vyslané senzorem. Ta je předána dále aplikaci a zobrazena na webovém rozhraní. Pokud zpráva ke klientovi nedorazí, nebo je zvoleno zařízení bez teplotního čidla je jako hodnota teploty zobrazeno -99. tato úprava brání zamrznutí aplikace. Pro čtení teploty by bylo možné použít i plnohodnotného MQTT klienta. To by si však vyžádalo spouštění v samostatném vlákně pro každý připojený senzor [26].

### 4.4.5 Implementace do webového rozhraní

Pro správné fungování webové aplikace bylo nutné upravit některé zdrojové soubory, starající se o správu `Socket.IO` kanálu a webové rozhraní zobrazující nastavení celého systému CraftBeerPi. Předně se jedná o fázi nastavení aktorů a teplotních čidel.

Zde byly vytvořeny nové položky pro novou možnost připojení hardwaru. Informace pro framework Flask jsou předávány souborem `__init__.py`. Zde došlo k přidání odkazů na jednotlivé nově vytvořené soubory. Po těchto úpravách a spuštění serverové aplikace se provedené změny projevily přímo ve webovém rozhraní a to přidáním položky ESP Easy v sekci nastavení hardwaru (viz Obr. 4.7).

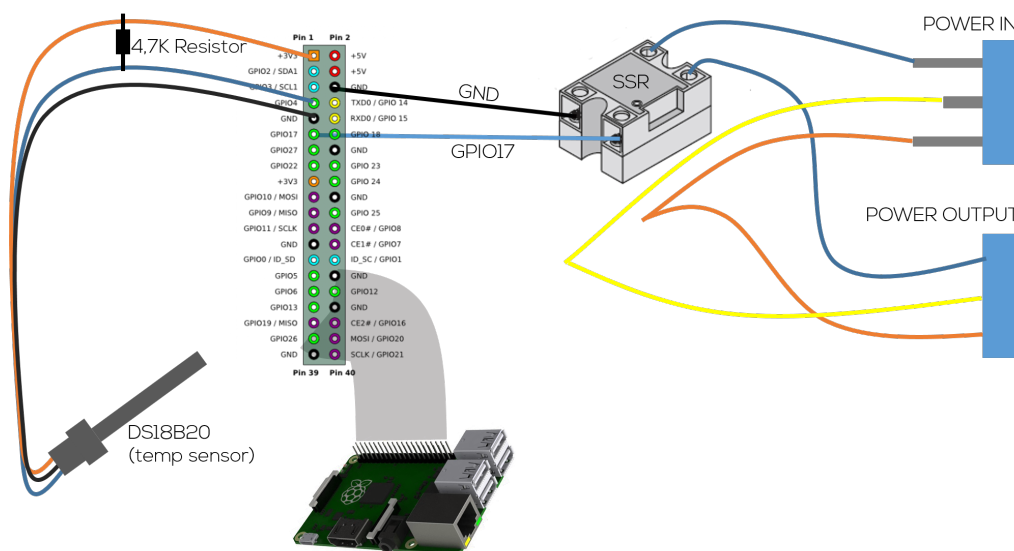


Obr. 4.7: Úvodní nastavení systému po úpravách



## 5 PŘIPOJENÍ SPÍNAČŮ A SENZORŮ K RASPBERRY PI

Program CraftBeerPi umožňuje ovládání připojených zařízení a zpětnou vazbu sledovaných hodnot ihned po spuštění serverové aplikace. Primárním ovládacím I/O rozhraním jsou porty GPIO. V nastavení webové aplikace musí být zvolen způsob ovládání přes GPIO a snímání teploty přes 1-Wire sběrnici. Poté v nastavení hardwaru je u požadovaného aktoru zvolen vstupně/výstupní port z rozbalovacího seznamu. Po uložení se takto nastavená položka zařízení objeví na úvodní obrazovce a je možné její aktivováním okamžitě ovládat příslušný GPIO port. 1-Wire sběrnice je automaticky dostupná na portu GPIO4 a při přiřazování teplotních čidel automaticky načte logické adresy všech připojených. Připojení spínačů se provádí přímo na korespondující port Raspberry Pi. Ten je po vydání příkazu nastaven na hodnotu log. 0 nebo 1, v závislosti na nastavení aplikace (sepnuto/rozepnuto, inverzní logika... ). Takto vyslaný signál může být zpracován spínacím obvodem, navrženou uživatelem (relé, SSR). Pro zajištění správného fungování 1-Wire sběrnice musí být vstupní pin GPIO4 připojen pull-up rezistorem o hodnotě 4,7 k $\Omega$  k napájecímu napětí 3,3 V. Na tuto sběrnici jsou poté připojeny datové výstupy jednotlivých čidel (např. DS18B20). Může také sloužit k parazitnímu napájení připojených zařízení[27].



Obr. 5.1: Nejjednodušší zapojení hardwaru CraftBeerPi [27]

Dalším možným způsobem ovládání hardwaru je rozhraní USB. Ovládaným zařízením jsou v tomto případě USB zásuvky Gembird. Po připojení zásuvek do USB portu se aplikace sama postará o nalezení všech potřebných parametrů a v nastavení webového rozhraní se zobrazí všechny načtené. Přiřazení je provedeno stejně jako

v případě ovládání přes GPIO. Po přiřazení lze jednotlivé zásuvky okamžitě ovládat. Při výběru teplotních čidel lze zvolit možnost USB Thermometer, která umožní získávání teploty ze všech teplotních čidel, připojených na USB porty Raspberry Pi.

## 5.1 Vytváření bezdrátového spojení

Jelikož zvolené bezdrátové spínače Sonoff komunikují přes bezdrátovou lokální síť, byl pro vytvoření komunikačního kanálu zvolen standard IEEE 802.11g. Podporovaný standard 802.11b byl zavržen z důvodu nízké přenosové rychlosti a standard 802.11n některá zařízení ESP8266 nemusí podporovat. Raspberry Pi slouží jako přístupový bod (AP) a moduly ESP8266 ve spínačích Sonoff jsou k tomuto bodu připojeny jako klienti.

Pro vytvoření přístupového bodu musí být Raspberry Pi vybaveno síťovým WiFi zařízením a řídicí aplikací. Použité Raspberry Pi model 3 B WiFi rozhraní již obsahuje. V základním nastavení je však pouze schopné se k AP připojovat, ne je vytvářet. K vytvoření AP slouží aplikace `hostapd`. Ta umožňuje i jejich správu, zabezpečení a řízení vysílacích parametrů. Do OS Raspbian se instaluje za použití správce balíčků s právy superuživatele (např. `sudo apt-get hostapd`). Pro správnou funkci musí mít korespondující bezdrátové rozhraní (v tomto případě `wlan0`) nastavenou statickou IP adresu. Jedním z možných způsobů provedení je úprava konfiguračního souboru s nastavením síťových rozhraní (`/etc/network/interfaces`).

Dalším bodem konfigurace bezdrátového přístupového bodu je nastavení aplikace `hostapd`. Veškeré nastavení je provedeno přes konfigurační soubor `hostapd.conf`, který je umístěn v adresáři `/etc/hostapd`. Soubor lze editovat jakýmkoliv vhodným textovým editorem (např. `nano`, `VIM`). Nejdůležitějším nastavovaným parametrem je SSID (Service Set Identifier – Identifikátor bezdrátové sítě). Dalšími možnými parametry je zabezpečení, vysílací kanál a mnohé další. Veškeré možnosti nastavení lze vyčíst z rozsáhlé dokumentace k tomuto programu. Pro tuto bakalářskou práci bylo zvoleno SSID `Control_AP`, zabezpečení WPA2-PSK a vysílání na šestém kanálu ve standardu 802.11g. Statická IP adresa je 10.0.1.2. Nastavení je dokončeno spuštěním démona `hostapd`. Pro správnou funkci je ale nutné do nastavení démona ještě připojit informaci, kde se nachází konfigurační soubor `hostapd.conf`. Konfiguračním souborem démona je `/etc/default/hostapd` [28].

V bakalářské práci bylo posledním krokem nastavení jednoduchého DHCP (Dynamic Host Control Protocol) serveru, který se bude starat o přidělování IP adres všem připojeným zařízením. Jako server byla zvolena aplikace `isc-dhcp-server`. Jedná se o jednoduchý, volně dostupný DHCP server. Konfigurační soubor DHCP serveru je umístěn ve složce `/etc/dhcp`. Nastavení je uvedeno ve výpisu 5.1 [29].

Výpis 5.1: Použitá konfigurace DHCP serveru

```
1 ddns-update-style none;
2
3 default-lease-time 600;
4 max-lease-time 7200;
5
6 authoritative;
7
8 log-facility local7;
9
10 subnet 10.0.1.0 netmask 255.255.255.0 {
11     range 10.0.1.10 10.0.1.254;
12     option broadcast-address 10.0.1.255;
13     option routers 10.0.1.1;
14     default-lease-time 600;
15     max-lease-time 7200;
16     option domain-name "Control_AP";
17     option domain-name-servers 8.8.8.8, 8.8.4.4;
18 }
```

## 5.2 Bezdrátový mikrokontroler ESP8266

Hlavním prvkem spínačů Sonoff je bezdrátový mikrokontroler ESP8266 (na Obr. 5.2). Jedná se o zařízení čínské firmy Espressif Systems, poprvé uvedený na trh v roce 2014. Obsahuje 32 bitový procesor s redukovanou instrukční sadou Tensilica Xtensa L106, běžící na frekvenci 80 MHz s 64 kB vnitřní pamětí RAM pro instrukce, 96 kB RAM pro data, externí FLASH pamětí až do velikosti 5 MB. Podporuje WiFi standardy IEEE 802.11 b/g/n, 16 GPIO portů a sběrnice jako I<sup>2</sup>C, SPI, UART, atd. Díky své nízké ceně se stal velmi populárním a vytvořil si širokou komunitu uživatelů. Programování je umožněno přes UART sběrnici a díky hotovým knihovnám jej lze provést například z Arduino IDE nebo originálního SDK Espressif. Existuje i řada již hotových programů uzpůsobených pro programování například přes USB/UART převodník nebo přes výrobky Arduino. Do programovacího režimu se mikrokontroler přepne po spojení portu GPIO0 s nulou a následném připojení napájení a datové sběrnice. Zařízení Sonoff mají UART sběrnici vždy vyvedenou na DPS a k portu GPIO0 je připojeno spínací tlačítko [30].



Obr. 5.2: Bezdrátový mikrokontroler ESP8266

### 5.2.1 Výběr vhodného řídicího programu pro bezdrátový mikrokontroler ESP8266

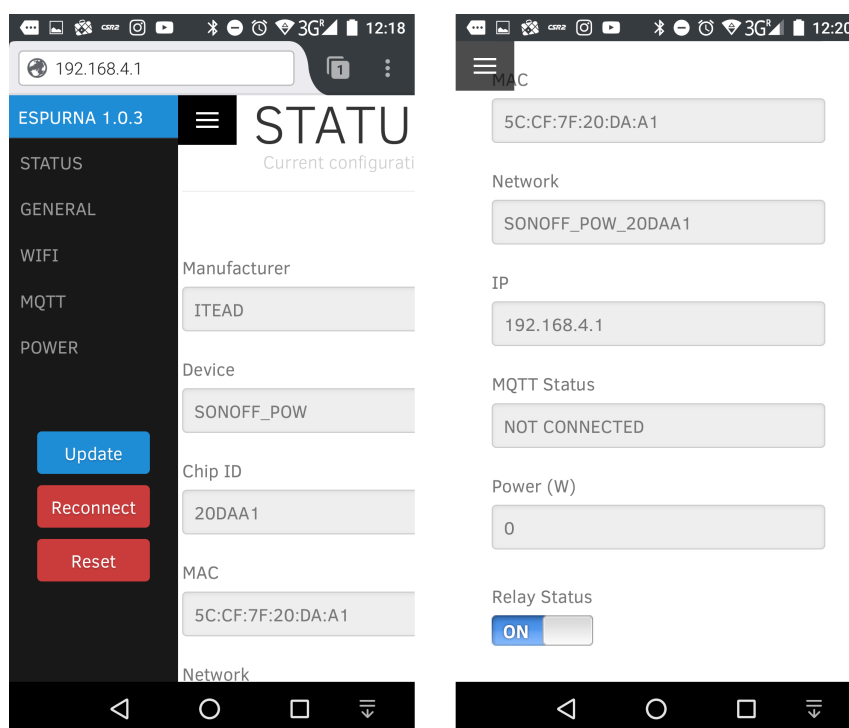
Originální software spínačů Sonoff nevyhovoval požadavkům bakalářské práce. To bylo dáno tím, že je kód majetkem společnosti Itead a celkové řešení aplikace bylo pro implementaci do CraftBeerPi nepoužitelné. Software bezdrátového modulu nekomunikuje s řídicí aplikací přímo, ale přes cloudový server společnosti a řídicí aplikace je primárně vyvíjena pro mobilní zařízení s operačními systémy Apple IOS nebo Android. Vytvářet nastavbu pro komunikaci ze zařízení Raspberry Pi by bylo velmi složité. Autor se tedy rozhodl o vymazání originálního softwaru a jeho náhradu vhodnějším. Nevýhodou této úpravy je ztráta záruky spínače Sonoff.

Nejzákladnější variantou programování, je již zmíněné Arduino IDE. Po připojení modulu k programátoru lze nahrávat vlastní kód psaný převážně v jazyce C. Existuje mnoho volně dostupných knihoven pro programování, připojení a ovládání periférií, WiFi spojení a odesílání dat. Výhodou vlastního programu je jednoduchost řešení a psaní aplikace přímo na míru požadovaným vlastnostem systému. Při větších nárocích na přenášená data a složitost systému však roste i náročnost na uživatelské znalosti programování a lépe nám poslouží již hotový software s otevřeným zdrojovým kódem.

#### Řídicí program ESPurna

Jedním z příkladů již hotového firmwaru pro ESP8266 je ESPurna (na Obr. 5.3). Jedná se o projekt s otevřeným zdrojovým kódem, který vznikl primárně pro nahrazení originálního firmwaru spínačů Sonoff. Tento program umí zařízení připojit až k pěti bezdrátovým sítím najednou a dokáže tak zabezpečit ovládání z několika různých zdrojů. Pro připojení periferních zařízení (čidla, senzory) je k dispozici několik

základních knihoven. Díky rozrůstající se uživatelské základně dochází k postupné implementaci senzorů mnoha dalších výrobců. Pro ovládání vstupů a výstupů je použit primárně protokol MQTT. Kromě toho umožňuje integraci do mnoha IoT řešení s otevřeným zdrojovým kódem. Ke konfiguraci slouží jednoduché integrované webové rozhraní, běžící na interním asynchronním HTTP serveru. Pro zobrazení dat v reálném čase v uživatelském prohlížeči používá firmware ESPurna Websockets. Zajímavým doplňkem je možnost integrace do systému, který je ovládán hlasovým asistentem Alexa od firmy Amazon[31].



Obr. 5.3: Řídící program ESPurna pro ESP8266

### Řídící program ESP Easy

Firmware ESP Easy nabízí kompletní řešení pro integraci jakéhokoliv zařízení s bezdrátovým modulem ESP8266 do jednoduchých, ale i rozsáhlých IoT projektů. K nastavení slouží velmi přehledně zpracované webové rozhraní na integrovaném HTTP serveru. Umožňuje implementaci do několika IoT projektů s otevřeným zdrojovým kódem převážně pomocí protokolu MQTT. Je zde však možnost vytvořit si vlastní pravidla pro odesílání MQTT zpráv. Dalším způsobem ovládání je pomocí protokolu HTTP. Hlavní menu je členěno do pěti záložek. První poskytuje obecné informace o zařízení. Jsou zde položky jako název, IP adresa, síla přijímaného signálu a další. Druhá záložka **Config** slouží ke konfiguraci zařízení. Zde je možné editovat název,

přiřadit adresy. . . Nachází se zde i menu pro editaci WiFi spojení, IP adres a portů řídicí aplikace zařízení, na které bude ESP Easy vysílat svá data. Menu **Hardware** umožňuje nastavit vstupně výstupní porty po spuštění systému na určitou logickou hodnotu. Nachází se zde konfigurace SPI rozhraní a přiřazení LED diody, která bude indikovat status zařízení. V menu **Devices** (Obr. 5.4) dochází k přiřazení jednotlivých vysílacích parametrů, konfiguraci připojených senzorů a čidel (např. DS18B20) a k přiřazení vstupních portů. V menu **Tools** můžeme nalézt nástroje pro správu systému od Restartu zařízení, přes správu konfigurace až po pokročilá nastavení (**Advanced**), sloužící například pro nastavení komunikace se Syslog serverem, skenování I<sup>2</sup>C sběrnice a dalších. Programování se provádí pomocí spustitelného skriptu, který uživatele provede nastavením, potřebným pro zapsání řídicího programu do FLASH paměti zařízení (nejprve je vybrán komunikační port, poté zvolena velikost vnitřní paměti a nakonec dojde k samotnému zápisu). Instalační program je součástí archivu se zdrojovými soubory ESP Easy [32].

Main Config Hardware **Devices** Tools

< >	Task	Device	Name	Port	IDX/Variable	GPIO	Values
Edit	1	Temperature & Humidity - SI7021	SI7021		257	GPIO-4 GPIO-5	Temperature: 18.17 Humidity: 39.00
Edit	2	Temperature & Pressure - BMP085	BMP085		243	GPIO-4 GPIO-5	Temperature: 17.31 Pressure: 1017.78
Edit	3	Luminosity - BH1750	BH1750		244	GPIO-4 GPIO-5	Lux: 194.17
Edit	4	Temperature & Humidity & Pressure - BME280	BME280		254	GPIO-4 GPIO-5	Temperature: 18.08 Humidity: 40.22 Pressure: 1019.48

Obr. 5.4: Příklad připojených zařízení přes ESP Easy

### 5.2.2 Instalace a nastavení řídicího programu ESP Easy

Pro implementaci ovládacího programu na vestavěný systém ESP8266 spínačů Sonoff autor použil originální skript, distribuovaný společně s programem ESP Easy. Pro zavedení do FLASH paměti zařízení byla využita sériová linka, dostupná na DPS a USB/TTL převodník. Před připojením je nutné modul ESP8266 přepnout do programovacího režimu přivedením logické nuly na port GPIO0 před připojením napájení a spuštění programovací aplikace. Po zvolení komunikačního portu dochází ke specifikaci velikosti vnitřní paměti spínače Sonoff (1024 kb) a program je automaticky zaveden. Restartem je nový řídicí program spuštěn. V bezprostředním okolí spínače je možné detekovat novou bezdrátovou síť **ESP\_0**. Pro zabezpečení je použito heslo **configesp**. Po připojení se zobrazí automatický dialog ve kterém dojde

k přesměrování komunikace na vybranou dostupnou WiFi síť. V přehledném menu lze vidět seznam SSID dostupných sítí, možnost vyplnit vlastní SSID (např. pokud je síť skrytá) a pole pro zadání hesla. Po potvrzení se zařízení automaticky odhlásí a připojí do zvolené bezdrátové LAN.

Main Config Hardware Devices Tools

Main Settings	
Name:	Sonoff01
Admin Password:	
SSID:	ControlAP
WPA Key:	raspberry
WPA AP Mode Key:	configesp
Unit nr:	1
Protocol:	OpenHAB MQTT ?
Locate Controller:	Use IP address
Controller IP:	10.0.1.2
Controller Port:	1883

Obr. 5.5: Nastavení psínače Sonoff01

K další konfiguraci dochází z bezdrátové LAN, ke které je spínač přihlášen. Nastavení se provádí přes internetový prohlížeč, takže pokud konfigurace probíhá přímo z Raspberry, je potřeba pracovat v grafickém rozhraní (připojením periférií k zařízení nebo přes virtuální grafický přístup – VNC). Modul ESP8266 získává svoji první IP adresu z DHCP serveru. Jednoduchým způsobem zjištění přiřazené adresy je ping na několik IP adres od počátku přiděleného rozsahu. Elegantnějším řešením je však přidělené adresy vyčíst přímo z DHCP serveru (soubor `/var/lib/dhcpd/dhcpd.leases`). Poté je možné připojit se na spínač, kde lze okamžitě vidět zobrazenou úvodní obrazovku zařízení. Na ní lze vyčíst parametry jako System uptime, přidělenou IP adresu, MAC adresu, identifikační číslo ESP čipu a další. Pro tuto práci však byla nejdůležitější záložka **Config**, kde dochází k nastavení názvu zařízení (v této práci Sonoff01 až 04), způsobu předávání informací (protokol MQTT, HTTP nebo Telnet). V případě nastavení protokolu MQTT dochází ještě ke specifikaci verze a nastavení IP adresy Brokeru (v této práci adresa Raspberry Pi – 10.0.1.2). Z této karty je také možno přidělit statické nastavení sítě (Obr. 5.5). V záložce **Hardware** lze přiřadit jeden GPIO výstup pro indikaci stavu zařízení. Spínače Sonoff mají indikační LED na portu GPIO13. V menu **Devices**

dochází k přiřazení senzorů. V této práci je použito teplotní čidlo DS18B20 a čidlo teploty/vlhkosti AM2301. 1-Wire sběrnice je připojena na port GPIO14 a po jejím aktivování jsou automaticky načteny adresy připojených zařízení. Je zde nastaven název proměnné (temp pro teplotu, hydro pro vlhkost) a upraven interval odesílání zpráv na 1 sekundu (Obr. 5.6). Posledním změněným parametrem je v menu **Advanced** nastavení tvaru zprávy na `/%Sysname%/%Valname%/`. Toto nastavení spíše koresponduje s nastavením klientské aplikace CraftBeerPi, která zajišťuje komunikaci přes protokol MQTT. Po restartování Sonoffu, Spuštění MQTT serveru mosquitto a spuštění aplikace CRAFTBeerPi, je uživatel vyzván ke zvolení způsobu spínání a odečítání teploty. Pokud je zvoleno nastavení nově přidávaných položek ESP Easy a 1Wire ESP Easy dojde k automatickému načtení všech připojených MQTT zařízení a jejich zobrazení v uživatelském rozhraní aplikace.

Task Settings	Value
Device:	Temperature - DS18b20 ?
Name:	temp
Delay:	1
IDX / Var:	1
1st GPIO:	GPIO-14
Device Nr:	1 ROM: 28-ff-f6-6d-85-16-3-8c
Send Data:	<input checked="" type="checkbox"/>

Optional Settings	Value
Formula temp:	Decimals: 2 ?
Value Name 1:	temp

Close Submit

Obr. 5.6: Nastavení připojeného senzoru DS18B20

## 5.3 Shrnutí a rekapitulace instalace a úprav systému CraftBeerPi

Zprovoznění aplikace CraftBeerPi začíná stažením bitového obrazu operačního systému Raspbian. Poté dochází k instalaci verzovacího systému Git a stažení zdrojových souborů CraftBeerPi. Tento program je poté instalován pomocí přiloženého skriptu `install.sh`. V této fázi je připravena distribuce systému CraftBeerPi k

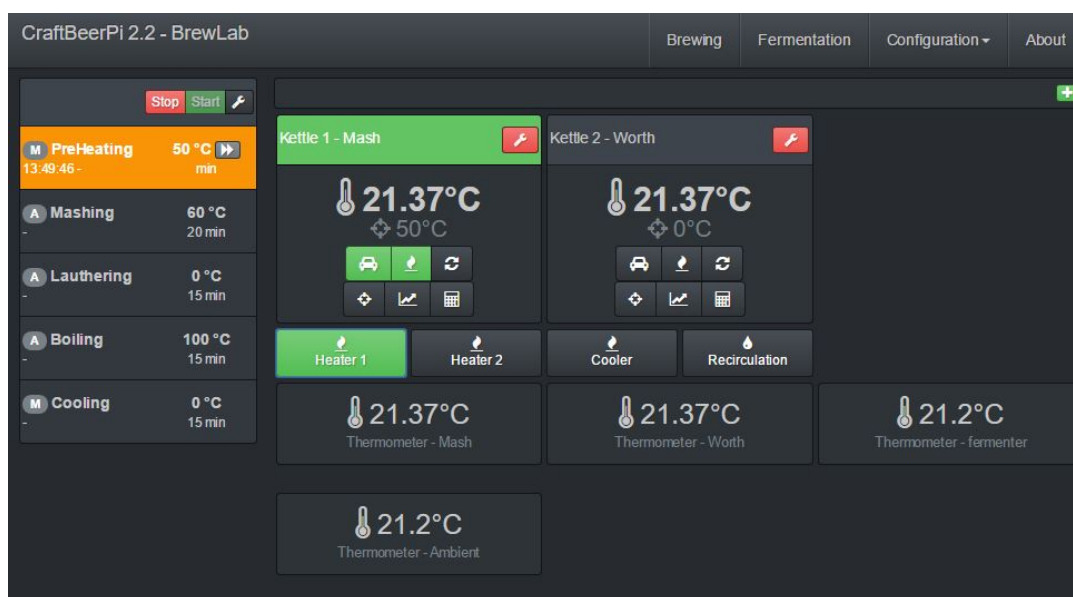


použití. Při požadavku na bezdrátovou komunikaci s ovládaným hardwarem jsou tři možná řešení. První a nejjednodušší je vypustit instalaci ze systému **Git** a použít zdrojové soubory z příloženého datového nosiče této práce. Poté může být spuštěn upravený instalační skript **install.sh**, který do operačního systému automaticky stáhne všechny potřebné závislosti, včetně souborů pro bezdrátové řízení. Další možností je kopírování upravených souborů z přílohy této práce do již existující instalace **CraftBeerPi**. V tomto případě se uživatel neobejde bez dodatečných zásahů v podobě stažení aplikací potřebných pro funkci systému (MQTT broker **mosquitto**, Eclipse **paho-mqtt**). Poslední možností je vytvoření vlastních zdrojových souborů podle této bakalářské práce. Tuto možnost je výhodné využít pokud je vyžadována úplná kontrola nad nástavbou pro bezdrátovou komunikaci programu **CraftBeerPi** a měli by se do ní pouštět pouze zkušení programátoři. Instalace dále pokračuje přeprogramováním spínačů Sonoff (nebo jakéhokoliv jiného zařízení s modulem ESP8266) novým řídicím programem **ESP Easy**. Poté lze přikročit ke kroku připojení ovládaných zařízení a MQTT brokera do jedné LAN sítě. Konfigurace použitá pro počítač Raspberry Pi bude záviset na podobě komunikační sítě. V případě využití již existující bezdrátové sítě lze v této nalézt nová připojená zařízení Sonoff a provést jejich konfiguraci. V této práci však byla pro bezdrátová zařízení využita oddělená bezdrátová síť **Control\_AP** což si vyžádalo instalaci a nastavení zprostředkovávající aplikace **hostapd** a DHCP klienta. Po těchto krocích je řídicí část systému plně připravena k použití.

Pořízení kompletního vybavení domácí varny přibližně podle návrhu autora, který je nastíněn v kapitole 3.3 je velmi nákladné. Cena elektroniky se pohybuje v řádu několika tisíc Kč. Mnohem nákladnější je však pořízení varného systému. V České republice se momentálně nenachází žádný distributor varných nádob pro domácí minipivovary a ze zahraničních obchodů je patrné, že cena jedné varné nádoby o objemu do 20 litrů se pohybuje okolo 5000 Kč. Náklady na zakázkovou výrobu by však pravděpodobně byly ještě mnohem větší. Autor cenu varné soupravy v nastíněném uspořádání, tedy dvě varné nádoby, jedna nádoba fermentační a jejich vzájemné propojení, odhaduje kolem 15000 Kč. Cena kompletního vybavení varny je tedy odhadnuta na několik desítek tisíc Kč.

## 6 NÁVOD K POUŽITÍ AUTOREM UPRAVE- NÉHO SYSTÉMU CRAFTBEERPI

Po dokončení úprav a nastavení systému pro vzdálený monitoring a řízení procesu výroby piva je možné začít aplikaci CraftBeerPi plně využívat. Finální nastavení spínačů a senzorů je vždy přizpůsobeno obsluhovaným zařízením. Tedy počtu varných nádob, teplotních čidel a aktorů. Demonstrace vzorového nastavení je provedena simulováním zařízení malé domácí varny v uspořádání RIMS, která obsahuje dvě varné nádoby a jednu nádobu fermentační. Ovládání je provedeno bezdrátově dle kapitoly 3.3 a sestává z jednodeskového počítače Raspberry Pi model 3B s nainstalovaným potřebným softwarovým vybavením, 4 bezdrátových spínačů Sonoff TH16 s řídicím programem ESP Easy, tří teplotních čidel DS18B20 a jednoho čidla teploty a vlhkosti AM2031. Veškeré nastavení hardwaru a softwaru je provedeno dle předchozích kapitol.



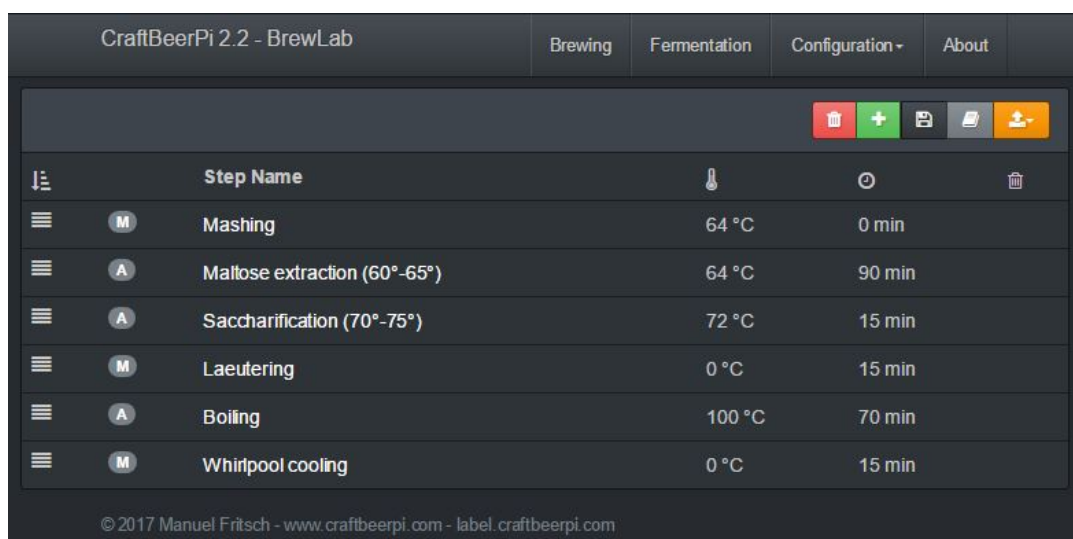
Obr. 6.1: Příklad nastavení vzorového systému pro vzdálený monitoring

Po spuštění aplikace CraftBeerPi je uskutečněno prvotní nastavení systému. Jako pracovní název simulovaného minipivovaru bylo zvoleno jméno BrewLab. V systému jsou dále nastaveny dva vyhřívací elementy Heater 1 a 2, jedno spínané chladicí zařízení (Cooler) a jedna přečerpávací pumpa (Pump), která bude zajišťovat recirkulaci a přečerpání obsahu mezi nádobami. Dále jsou přiřazena teplotní čidla k jednotlivým nádobám. Senzor AM2031 slouží k zobrazení teploty a vlhkosti v prostorách varny a jeho vysílané hodnoty jsou pouze informativní. V systému jsou dále nastaveny dvě varné nádoby (Kettle 1 – Mash a Kettle 2 – Worth) a jedna nádoba fermentační.

tační (Fermenter) Nakonec dochází k přiřazení hardwaru. Změna úvodní obrazovky systému po nastavení je zobrazena na Obr. 6.1.

## 6.1 Správa receptů

Nastavení automatického sledu kroků vaření ve formě receptu lze vidět na levé straně úvodní obrazovky. Po kliknutí na ikonu ve tvaru klíče se otevře Správce receptů, ve kterém lze jednotlivé kroky vytvářet, upravovat a ukládat do databáze receptů. Recepty lze importovat do systému například z programu Kleiner Brauhelfer nebo je exportovat a nahrávat z xml souborů. Ovládací prvky jsou zobrazeny v pravé horní části panelu správce receptů (viz Obr. 6.2).

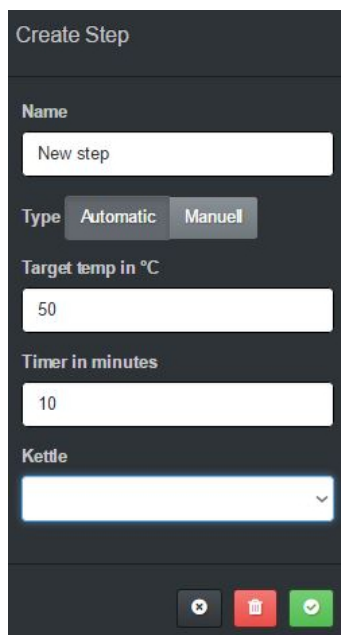


	Step Name	🌡️	⌚	🗑️
☰	M M Mashing	64 °C	0 min	
☰	A A Maltose extraction (60°-65°)	64 °C	90 min	
☰	A A Saccharification (70°-75°)	72 °C	15 min	
☰	M M Laeuterung	0 °C	15 min	
☰	A A Boiling	100 °C	70 min	
☰	M M Whirlpool cooling	0 °C	15 min	

Obr. 6.2: Správce receptů programu CraftBeerpi

Kliknutím na tlačítko plus lze přidat nový krok receptu na konec seznamu předchozích kroků. Otevře se okno (viz Obr. 6.3 ve kterém je uživatel vyzván k zadání několika důležitých informací. Prvním nastaveným bodem je název kroku. Ten musí být v každém receptu unikátní a je vhodné do něj umístit i základní informace (např. teplotu, co je krokem sledováno a další). Poté dochází ke zvolení typu kroku. Tedy jestli bude po jeho skončení automaticky vykonán krok další, nebo je nutný uživatelský zásah a na další krok bude přepnuto manuálně. Poté dochází ke zvolení požadované teploty, která má být držena a nastavení časovače, jak dlouho se bude krok vykonávat. Délka kroku se udává v minutách a je důležitá především pro automatický režim. Při jakémkoliv časovém intervalu dochází ke spuštění časovače až po dosažení nastavené teploty. Pokud je čas nulový, je na další krok přepnuto ihned po dosažení teploty. Pokud je pole ponecháno prázdné na další krok nebude přepnuto

bez uživatelského zásahu. V posledním poli nastavení kroku je z rozbalovacího seznamu zvolena nádoba, pro kterou dochází k nastavení parametrů podle příslušného kroku. Po potvrzení nastavení je krok zobrazen na konci seznamu. Případná úprava se provádí kliknutím na název kroku. V seznamu jsou patrné všechny nastavené parametry. Nalevo od názvu lze vidět typ kroku (Automatický/Manuální), napravo poté nastavenou teplotu, čas a nádobu. Jednotlivé kroky lze po zakliknutí přesunovat a měnit tak jejich pořadí.



Obr. 6.3: Vytvoření nového kroku v seznamu receptů

## 6.2 Zahájení a průběh procesu vaření piva

Po nahrání receptu lze přikročit k samotnému vaření piva. Je však vhodné nejdříve provést kontrolu nastavených parametrů a vizuální kontrolu vybavení varny. Je dobré se též ujistit o správném nastavení varných nádob (především nastavení logiky, bez které nebude provedeno spínání) a jejich přepnutí do auto automatického režimu spínání (ikona ve tvaru automobilu na úvodní obrazovce). Po zavedení surovin je spuštěn recept tlačítkem **Start** na úvodní obrazovce. Dochází ke spuštění prvního kroku (zpravidla předehřívání systému s nastaveným nulový časem). V seznamu lze vidět změnu barvy pozadí kroku ze šedé na oranžovou, která indikuje status receptu jako aktivní. Po dosažení požadované teploty dojde ke spuštění časovače a po uplynutí časového intervalu k přepnutí na další krok nebo vyčkání na zásah uživatele. Další aktivní krok změní svou barvu pozadí na oranžovou a pozadí předchozího

vykonaného kroku se zabarví modře. Přepínání kroků je indikováno i akusticky, pokud je k systému připojen reproduktor na port GPIO16. Recept je postupně vykonáván až do posledního kroku, ve kterém dojde k ukončení vaření a informování uživatele. Celý automatický postup lze kdykoliv zastavit kliknutím na tlačítko **Stop**.

## ZÁVĚR

Hlavním výstupem této bakalářské práce je plně funkční aplikace systému pro vzdálený monitoring a řízení procesu výroby piva. Jejím základem je volně dostupný program CraftBeerPi, který byl autorem vhodně upraven. Díky těmto úpravám došlo ke stabilizaci celé aplikace, snížení doby odezvy grafického uživatelského rozhraní, ale především přidání možnosti ovládat připojený hardware bezdrátově.

Postup práce probíhal od zjištění vlastností dostupných systémů pro vzdálený monitoring s otevřeným zdrojovým kódem, přes implementaci zvoleného systému na testovací elektronické vybavení malé domácí varny, až po samotnou úpravu serverové aplikace a její testování s bezdrátovými spínači Sonoff TH16. Implementace aplikace CraftBeerPi a její úprava je v práci detailně popsána. Po prozkoumání možností již hotového systému byl vytvořen jednoduchý uživatelský návod na obsluhu. Práce tedy může sloužit jako příručka pro nadšence se zájmem vyzkoušet si vaření piva v domácích podmínkách.

Aplikaci CraftBeerPi hodnotí autor jako velmi obstojnou v porovnání s ostatními dostupnými programy. Kód psaný v programovacím jazyce Python je zapsán velmi přehledně, bohužel však s absencí komentářů. Složitost celé aplikace si však v případě jakékoliv úpravy kódu vynutila zdlouhavé pročítání zdrojových souborů a testování před kompletací. S programovacím jazykem Python se autor setkal poprvé, což také těž znesnadnilo práci, nicméně tím byla získána spousta cenných informací. Velmi přínosná byla též práce s IoT protokolem MQTT. Fenomén Internetu věcí je záležitost relativně nová, velmi rychle se rozvíjející a jakékoliv zkušenosti s touto oblastí budou velmi užitečné.

Práce na tomto projektu (pojmenován BrewLab) bude pravděpodobně do budoucna pokračovat. Jako další možnou úpravu systému autor vidí přidání možností ukládat v grafickém rozhraní aktory do skupin, které by se daly ovládat najednou (např. pro spínání třífázových obvodů). Pravděpodobně bude vytvořen i nový instalační skript, který uživatele provede procesem instalace nástavby bezdrátového řízení na již instalovanou aplikaci CraftBeerPi. Úpravy systému budou též nutné při změně verzí výchozí aplikace (momentálně je připravována verze CraftBeerPi 3.0) tak, aby uživatelé mohli nástavbu bez problémů dále používat.

## LITERATURA

- [1] *Stručný postup výroby piva*[online]. Pivní klenoty 2014[cit. 2016-10-12] Dostupné z URL: <<http://www.pivniklenoty.cz/vse-o-pivu/prakticke-rady/strucny-postup-vyroby-piva/>>.
- [2] *Novoměstský pivovar v Praze*[online]. [cit. 2016-10-28] Dostupné z URL: <<http://www.prague.eu/cs/objekt/jidlo/588/novomestsky-pivovar>>.
- [3] *Domácí výroba piva*[online]. Výzkumný ústav pivovarský a sladařský 2012[cit. 2016-10-12] Dostupné z URL: <[http://www.beerresearch.cz/index.php?option=com\\_content&view=article&id=65&Itemid=165](http://www.beerresearch.cz/index.php?option=com_content&view=article&id=65&Itemid=165)>.
- [4] *Make Your Own Beer: 15 Great Homebrew Recipes To Try*[online]. [cit. 2016-10-28] Dostupné z URL: <<http://drinks.seriousseats.com/2013/02/best-homebrewing-recipes-how-to-brew-your-own-beer.html>>.
- [5] *The Internet of Things* [online]. IOT COUNCIL 2016[cit. 2016-10-15]. Dostupné z URL: <<http://www.theinternetofthings.eu/what-is-the-internet-of-things>>.
- [6] VODA, Zdeněk *Arduino Uno*LINUXSOFT[online]. 2014[cit. 2016-10-15]. ISSN 1801-3805. Dostupné z URL: <[http://www.linuxsoft.cz/article.php?id\\_article=1881](http://www.linuxsoft.cz/article.php?id_article=1881)>.
- [7] ČEVELA, Ladislav *Raspberry Pi 2 model B-představení, sestavení, instalace* OSLinuxEXPRESS[online]. 2015[cit. 2016-10-15]. ISSN 1801-3996. Dostupné z URL: <<http://http://www.linuxexpres.cz/hardware/raspberry-pi-2-model-b-predstaveni-sestaveni-instalace-os>>.
- [8] *Banana Pi: What is Banana Pi* [online]. 2011[cit. 2016-10-15]. Dostupné z URL: <<http://www.bananapi.org/p/product.html>>.
- [9] NEWTON, Jason. *Brewbot, Automating the Home brewing Process: Brewbot V2.0*[online]. Canada, 2013[cit. 2016-10-15]. Dostupné z URL: <<http://www.brewbot.ca/index.html>>.
- [10] *BrewPi*[online]. 2016[cit. 2016-10-15]. Dostupné z URL: <<http://www.brewpi.com>>.
- [11] *CraftBeerPi on GitHub* [online]. 2016[cit. 2016-10-15]. Dostupné z URL: <<https://github.com/Manuel83/craftbeerpi>>.

- [12] FRITSCH, Manuel. *CraftBeerPi 2.2* [online]. Paderborn 2015[cit.2016-10-15]. Dostupné z URL: <<http://www.craftbeerpi.com>>.
- [13] DÜLL, Sebastian. *HW – MashBerry* [online]. Veitsbronn 2015[cit.2016-10-15]. Dostupné z URL: <<http://sebastian-duell.de/img/mashberry/pictures/HW-all.JPG>>.
- [14] DÜLL, Sebastian. *MashBerry* [online]. Veitsbronn 2015[cit.2016-10-15]. Dostupné z URL: <<http://www.sansalinux.org/en/mashberry/index.html>>.
- [15] RONACHER, Armin. *Welcome to Flask* [online]. 2017 [cit.2017-3-1]. Dostupné z URL:<<http://flask.pocoo.org/docs/0.12/#user-s-guide>>.
- [16] *Raspberry Pi 2 Model B 1GB* [online]. 2016[cit.2016-10-25]. Dostupné z URL: <<http://rpishop.cz/raspberry-pi-pocitace/170-raspberry-pi-2-1024-mb-ram.html>>.
- [17] *DS18B20 teplotní čidlo* [online]. 2016[cit.2016-10-25]. Dostupné z URL: <<https://www.gme.cz/teplotni-cidlo-s-kabelem-2-metry-flajzar-ds18b20-teplotni-cidlo>>.
- [18] *SSR pro montáž na chladič* [online]. 2016[cit.2016-10-25]. Dostupné z URL: <<https://www.gme.cz/ssr-pro-montaz-na-chladic>>.
- [19] *Sonoff - WiFi Wireless Smart Switch For MQTT COAP Smart Home* [online]. 2015[cit.2017-3-1]. Dostupné z URL: <<https://www.itead.cc/sonoff-wifi-wireless-switch.html>>.
- [20] *Sonoff TH10/TH16: Temperature And Humidity Monitoring WiFi Smart Switch* [online]. 2015[cit.2017-3-1]. Dostupné z URL: <<https://www.itead.cc/sonoff-th.html>>.
- [21] *Raspbian* [online]. 2016[cit.2016-11-5]. Dostupné z URL:<<https://www.raspberrypi.org/downloads/raspbian/>>.
- [22] PEŠEK, Jiří. *Raspbian: Instalace Rasbianu na SD kartu z Windows* [online]. 2014[cit.2016-11-12]. Dostupné z URL:<<http://raspberrypi.cz/instalace-rasbianu-na-sd-kartu-z-windows/>>.
- [23] MALÝ, Martin *Protokol MQTT: komunikační standard pro IoTRoot.cz* [online]. 2016[cit.2017-11-12]. ISSN 1212-8309. Dostupné z URL:<<https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>>.



- [24] *Mosquitto: An Open Source MQTT v3.1/v3.1.1 Broker* eclipse.org[online]. 2016[cit. 2016-11-5]. Dostupné z URL: <<https://mosquitto.org/documentation/>>.
- [25] *Python Package Index: paho-mqtt 1.2.3* Python Software Foundation[online]. 2017[cit. 2016-11-5]. Dostupné z URL: <<https://pypi.python.org/pypi/paho-mqtt>>.
- [26] MONK, Simon *Programing the Raspberry Pi: getting started with Python*. New York McGraw-Hill. ISBN 00-718-0783-7.
- [27] FRITSCH, Manuel. *CraftBeerPi 2.2: Hardware Wiring*[online]. Paderborn 2015[cit. 2016-10-15]. Dostupné z URL: <<http://web.craftbeerpi.com/hardware/>>.
- [28] MALINEN, Jouni. *hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA-/WPA2/EAP/RADIUS Authenticator*[online]. 2013[cit. 2016-10-15]. Dostupné z URL: <<https://w1.fi/hostapd/>>.
- [29] *isc-dhcp-server* ubuntu documentation[online]. 2015[cit. 2016-10-15]. Dostupné z URL: <<https://help.ubuntu.com/community/isc-dhcp-server>>.
- [30] FOLTÝN, Petr. *Internet věcí - Internet of Things (IoT) a ESP8266* ITECO[online]. 2017[cit. 2016-10-15]. Dostupné z URL: <<https://www.arduinotech.cz/inpage/internet-veci/#>>.
- [31] PERÉZ, Xose. *ESPurna Firmware* Bitbucket[online]. Austin 2017[cit. 2016-10-15]. Dostupné z URL: <<https://bitbucket.org/xoseperez/espurna>>.
- [32] *ESPEasy* LetsControlIt[online]. 2017[cit. 2016-10-15]. Dostupné z URL: <<https://www.letscontrolit.com/wiki/index.php/ESPEasy>>.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AP	Access point – Přístupový bod
API	Application Programming Interface – rozhraní pro programování aplikací
BT	Bluetooth
DHCP	Dynamic Host Control Protocol
EPM	Extrakt původní mladiny
GPIO	General-purpose input/output
GSM	Groupe Spécial Mobile – Globální systém pro mobilní komunikaci
GUI	Graphical User Interface – Grafické uživatelské rozhraní
HDMI	High-Definition Multi-media Interface
HERMS	Heat Exchanged Recirculation Mash System – Recirkulační vytápěcí/rmutovací systém s teplotní výměnou
HTML	HyperText Markup Language
HTTP	HyperText Transfer protocol
I/O	Input/Output – vstupně výstupní
IoT	Internet of Things – Internet věcí
IP	Internet Protocol
LAN	Local Area Network – Lokální síť
MQTT	Message Queuing Telemetry Transport, MQ Telemetry Transport
QoS	Quality of Service – kvalita služeb
PC	Personal Computer – osobní počítač
PID	proportional–integral–derivative controller – regulátor s proporciální, integrační a derivační částí
PWM	Pulse Width Modulation – Pulzně šířková modulace
RAM	Random Access Memory – Paměť s náhodným přístupem

RIMS	Recirculation Infusion Mash System – Recirkulační Infúzní vystírací/rmutovací systém
SSID	Service Set Identifier – Identifikátor bezdrátové sítě
SSL	Secure Socket Layer – vrstva bezpečných socketů
SSR	Solid state relé
TSL	Transport Layer Security – Zabezpečení transportní vrstvy
UART	universal asynchronous receiver/transmitter
USB	Universal Serial Bus – Univerzální sériová sběrnice
VNC	Virtual Network Computing
WiFi	Standardy IEEE 802.11 pro připojení k bezdrátovým lokálním sítím

# SEZNAM PŘÍLOH

<b>A</b>	<b>Příloha – Vlastní zdrojové kódy</b>	<b>60</b>
A.1	Program pro získání seznamu připojených bezdrátových zařízení . . .	60
A.2	Program pro bezdrátové řízení výstupů přes MQTT protokol . . . .	62
A.3	Program pro čtení teploty z MQTT zpráv . . . . .	64
<b>B</b>	<b>Příloha – Obsah přiloženého CD</b>	<b>66</b>

## A PŘÍLOHA – VLASTNÍ ZDROJOVÉ KÓDY

### A.1 Program pro získání seznamu připojených bezdrátových zařízení

Výpis A.1: Soubor, spravující komunikaci přes protokol MQTT a generování seznamu připojených zařízení

```
1  #připojení MQTT knihovny
2  import paho.mqtt.client as mqtt
3
4  #Definice seznamů zařízení
5  switches = []
6  thermometers = []
7  hydrometers = []
8
9  #Volání funkce po připojení klienta
10 def on_connect(client, userdata, flags, rc):
11
12  #Subskribce v on_connect() zajistí nastavení tématu ihned
13  #po připojení
14      client.subscribe("#")
15
16  #Definovaná funkce, volaná po zachycení zprávy k tématu
17  def on_message(client, userdata, msg):
18      topic = []
19      typ = ''
20      name = ''
21
22  #Rozřezání říjatého řetězce na seznam znaků
23      topic = list(msg.topic)
24
25  #Úprava tvaru názvu zařízení
26      if str(topic[0]) == '/':
27          del topic[0]
28
29  #Selekce názvu zařízení z přijatého tématu zprávy
30      for i in range(0, int(topic.index('/'))):
31          name += str(topic[i])
32
33
```

```

34 #Selekce typu zařízení
35     for i in range(1+int(topic.index('/')), 1+int
36         (topic.index(str(topic[-1])))):
37         typ += str(topic[i])
38
39 #Přiřazení jména zařízení na konec příslušného seznamu
40     if name not in thermometers and typ == 'temp':
41         thermometers.append(name)
42     elif name not in hydrometers and typ == 'hydro':
43         hydrometers.append(name)
44     elif name not in switches:
45         switches.append(name)
46
47 #Funkce pro vrácení seznamu požadovaných zařízení
48 def getNames(device):
49     if device == 'thermometer':
50         return thermometers
51     elif device == 'hydrometer':
52         return hydrometers
53     else:
54         return switches
55
56 #Definice MQTT klienta a použitých zpětně volaných metod
57 client = mqtt.Client()
58 client.on_connect = on_connect
59 client.on_message = on_message
60
61 #připojení klienta k brokeru
62 client.connect("10.0.1.2", 1883, 60)
63
64 #Spuštění nekonečné smyčky klienta v samostatném vlákně
65 client.loop_start()

```

## A.2 Program pro bezdrátové řízení výstupů přes MQTT protokol

Výpis A.2: Soubor pro řízení vstupně výstupních portů zařízení ESP8266 bezdrátovým přenosem informací

```
1  #Připojení metody publish z knihovny paho.mqtt
2  import paho.mqtt.publish as publish
3
4  #Importování funkcí a tříd z CraftBeerPi
5  from brewapp import app
6  from brewapp.base.actor import ActorBase
7  from brewapp.base.model import *
8
9  #Propojení se souborem mqttHandler
10 from brewapp.base.mqttHandler import getNames
11
12 #Vytvoření třídy pro spínač
13 class espEasy(ActorBase):
14
15     def init(self):
16         self.state = True
17
18     def cleanup(self):
19         pass
20
21 #Funkce pro získání seznamu připojených zařízení
22     def getDevices(self):
23         names = getNames('switch')
24         return names
25
26 #Funkce pro sepnutí požadovaného výstupu
27     def switchON(self, device):
28         app.logger.info("ESP_ON" + str(device))
29
30 #Překlad jména z informací v databázi
31         switch_name=self.getConfigValue(device,"switch",None)
32
33         if switch_name is None:
34             app.logger.warning("SWITCH_NOT_FOUND_IN_CONFIG")
35         pass
```

```

36
37     if self.getConfigValue(device, "inverted", False) :
38
39         #Přidání záznamu o sepnutí do logu
40         app.logger.warning("SWITCH_ON- Inverted")
41
42         #Publikování MQTT zprávy
43         publish.single(str(switch_name)+"/cmd","GPIO,12,0",
44             hostname="10.0.1.2")
45     else:
46         app.logger.warning("SWITCH_ON- Not Inverted")
47         publish.single(str(switch_name)+"/cmd","GPIO,12,1",
48             hostname="10.0.1.2")
49     pass
50
51 #Definice funkce pro rozepnutí zvoleného výstupu
52     def switchOFF(self, device):
53         app.logger.info("Sonoff_OFF" + str(device))
54
55         switch_name=self.getConfigValue(device,"switch",None)
56
57         if switch_name is None:
58             app.logger.warning("SWITCH_NOT_FOUND_IN_CONFIG")
59             pass
60
61         if self.getConfigValue(device, "inverted", False) :
62             app.logger.warning("SWITCH_OFF- Inverted")
63             publish.single(str(switch_name)+"/cmd","GPIO,12,1",
64                 hostname="10.0.1.2")
65         else:
66             app.logger.warning("SWITCH_OFF- Not Inverted")
67             publish.single(str(switch_name)+"/cmd","GPIO,12,0",
68                 hostname="10.0.1.2")
69     pass

```



## A.3 Program pro čtení teploty z MQTT zpráv

Výpis A.3: Program na vyčtení informace o teplotě ze zachycené MQTT zprávy

```
1  #Připojení metody Subscribe z knihovny paho.mqtt
2  import paho.mqtt.subscribe as subscribe
3
4  #Propojení funkcí aplikace CraftBeerPi, Flask a SocketIO
5  import os, re
6  from subprocess import Popen, PIPE, call
7  from random import randint, uniform
8  from brewapp import app
9  from decimal import Decimal, ROUND_HALF_UP
10 from subprocess import call
11 import json
12 from flask import Blueprint, render_template, json, request
13 from brewapp.base.model import *
14 from brewapp import app, socketio
15
16 #Import funkce getNames() ze souboru mqttHandler
17 from brewapp.base.mqttHandler import getNames
18
19 #Definice třídy teplotního čidla
20 class espThermometer(object):
21
22     def init(self):
23         print "INIT"
24
25 #Funkce pro získání seznamu teplotních čidel
26     def getSensors(self):
27         names = getNames('thermometer')
28         return names
29
30 #Funkce pro čtení teploty z MQTT zprávy
31     def readTemp(self, tempSensorId):
32
33         #definice aplikace v testovacím režimu
34
35         temp_C = -99
36
37         if(tempSensorId == None or tempSensorId == ""):
38             return -23
```

```

39         if (app.testMode == True):
40             pipe = Popen(["cat","w1_slave"], stdout=PIPE)
41             result = pipe.communicate()[0]
42         else:
43 #V případě běžného režimu dochází k zachycení MQTT zprávy
44             msg = subscribe.simple(str(tempSensorId)+"/temp",
45                                     hostname="10.0.1.2", keepalive=5)
46
47 #Selekce teploty z těla zprávy a následná kontrola
48             temp = msg.payload
49             if (temp != None and temp != ""):
50                 temp_C = float(temp)
51
52 #Chybné čtení teploty ze zprávy
53             else:
54                 temp_C = -99 #bad temp reading
55
56         return temp_C

```

## B PŘÍLOHA – OBSAH PŘÍLOŽENÉHO CD

/	kořenový adresář přiloženého CD
└─ craftbeerpi	Zdrojové soubory originálního programu CraftBeerPi
└─ brewapp	Složka aplikace
└─ docs	
└─ upload	
└─ craftbeerpiboot	Spouštění po startu systému
└─ install.sh	Instalační skript
└─ LICENSE	
└─ manage.py	
└─ requirements.txt	Odkazy na závislosti programu
└─ runserver.py	Spuštění serverové aplikace
└─ w1 slave	
└─ datasheet	Katalogové listy některých použitých součástí
└─ img	Obrazy disků
└─ 2016-09-23-raspbian-jessie-lite.img	Raspbian OS
└─ boot.img	Obraz MicroSD paměťové karty
└─ pictures	Složka s obrázky a fotografiemi
└─ buttons GPIO.jpg	Popis GPIO výstupu
└─ dashboard full size.jpg	Webové rozhraní
└─ BrewLab	Zdrojové soubory upraveného programu CraftBeerPi
└─ brewapp	Složka aplikace
└─ craftbeerpiboot	Spouštění po startu systému
└─ install.sh	Instalační skript
└─ manage.py	
└─ requirements.txt	Odkazy na závislosti programu
└─ runserver.py	Spuštění serverové aplikace